

TEKNILLINEN KORKEAKOULU

Sähkö- ja tietoliikennetekniikan osasto

Jussi Sirkiä

HARJATTOMIEN TASAVIRTAMOOTTOREIDEN OHJAIN
LENNOKKIKÄYTTÖÖN

Diplomityö, joka on jätetty opinnäytteenä tarkistettavaksi
diplomi-insinöörin tutkintoa varten Espoossa 2.10.2006

Työn valvoja

Professori Jorma Kyyrä

Tekijä: Jussi Sirkiä	
Työn nimi: Harjattomien tasavirtamoottoreiden ohjain lennökkikäyttöön	
Päivämäärä: 2.10.2006	Sivumäärä: 60
Osasto: Sähkö- ja tietoliikennetekniikan osasto	
Professuuri: S-81 Tehoelektroniikka ja sähkökäytöt	
Työn valvoja: Professori Jorma Kyyrä	
<p>Radio-ohjattavissa lennokeissa on 1990-luvulta alkaen käytetty enenevässä määrin sähkömoottoreita vanhojen polttomoottoreiden sijasta. Sähkön edut ovat kiistattomat: hiljaisempi käyntiääni, olemattomat savu- ja öljypäästöt sekä halpa energia tekevät harrastuksesta huomattavasti mukavampaa sekä harrastajille että ympäristölle.</p> <p>2000-luvulla suosituimmaksi sähkömoottorityypiksi on noussut harjaton tasavirtamoottori, jonka vääntömomentti kokoon nähden on suuri ja hyötysuhde parempi kuin harjallisilla moottoreilla. Moottoria kommutoidaan elektronisesti, joten ohjauselektroniikka on monimutkaisempaa kuin perinteisillä tasavirtamoottoreilla. Toisaalta ohjauksen ajoitusta ym. parametreja voidaan säätää moottoriin koskematta.</p> <p>Tässä työssä suunnitellun moottorinohjaimen laitteisto kehitettiin laitteen käyttöolosuhteet huomioiden fyysisesti mahdollisimman pieniksi, ja erilaiset vaihtoehtoiset toteutustavat pyrittiin ottamaan huomioon. Käytännössä laitteiston toteutus on hyvin samankaltainen kaikissa lennökkiohjaimissa. Ainoa merkittävästi erilainen ratkaisu tässä työssä oli vaihejännitteiden kaksiporainen skaalaus ohjauksen pulssisuhteen mukaan.</p> <p>Ohjelmisto toteutettiin C-kielellä PIC-mikrokontrollerille. Pääpainotus ohjelman kirjoittamisessa oli sovelluksen mahdollisimman nopea suoritus, jotta moottorin ajoitukset pysyisivät hallinnassa vielä suurillakin kierrosnopeuksilla.</p> <p>Ohjaimen toiminta hiottiin ohjelmistoa parantamalla työn kuluessa kohtalaiseksi: Testimoottorit saatiin pyörimään silmämääräisesti arvioiden melko tasaisesti ja korkeillakin kierroksilla. Lähempi vaihejännitteiden tarkastelu oskilloskoopilla paljasti kuitenkin, että jaksosta jaksoon ajoituksen onnistuminen on vaihtelevaa, mikä laskee hyötysuhdetta.</p> <p>Jatkokehityksessä laitteen toiminnan kannalta ensisijaista on ohjelman suorituksen nopeuttaminen prosessorin tyyppiä vaihtamalla ja keskeytyksiä paremmin hyödyntämällä. Lisäksi tehokytkinten ohjausta tulisi parantaa siten, että laitteen minimikäyttöjännite saataisiin nykyistä alemmas.</p>	
Avainsanat:	
harjaton, moottori, ohjain, lennokki, radio-ohjaus	

Author: Jussi Sirkiä	
Name of the Thesis: A Brushless Direct Current Motor Controller for Radio Controlled Model Airplanes	
Date: 2.10.2006	Number of pages: 60
Department: Department of Electrical and Communications Engineering	
Professorship: S-81 Power Electronics and Electrical Drives	
Supervisor: Professori Jorma Kyyrä	
<p>Electric motors started to replace internal combustion engines in radio controlled model airplanes during the 90's. The benefits are significant: Electric motors are quiet, they don't produce oil or smoke and their energy is cheap. These facts make for a considerably nicer hobby both for the people involved and their surroundings.</p> <p>In recent years the brushless direct current motor has become the motor of choice due to it's large torque-to-size ratio and superior efficiency. The motor is commutated electronically, making the driving electronics more complicated, but on the other hand the commutation timing of the motor and other similar parameters can be changed without touching the motor.</p> <p>In this thesis, a controller for brushless motors was designed. The hardware was developed to be as small as possible, while taking in account any alternative ways of implementation. All in all the hardware is very similar in all airplane controllers, the only significant improvement in this work was the inclusion of a dual scaling for phase voltages according to the duty cycle used.</p> <p>The software was developed using C language for a PIC microcontroller. The main emphasis during programming was for the execution of the program to be as fast as possible, allowing to maintain correct timing even at very high rotational speeds.</p> <p>During development the operation of the controller was tuned by tweaking the software. In the end the test motors could be run fairly smoothly and even at high speeds. A closer inspection of the phase voltages with an oscilloscope, however, revealed quite a bit of variation in the timing, which leads to a reduction in the overall efficiency of the drive.</p> <p>For further development, the most critical point is to make program run-time execution even faster by changing the processor to a more suitable type and by making the most of available interrupts. In addition, the drivers of the power switches should be improved to allow for a much lower minimum input voltage.</p>	
Keywords:	
brushless, motor, controller, model airplane, radio control	

Alkulause

Tämä diplomityö pohjautuu kirjoittajan pitkäaikaiseen harrastukseen radio-ohjattavien lennokkien parissa. Varsinainen kehitystyö on tehty kirjoittajan päivätyön ohella PartnerTech Åbo Oy:n suunnitteluosastolla.

Haluan kiittää PartnerTech Åbo Oy:tä työn tekemisen mahdollisuudesta. Yrityksen tuki konkretisoitui työkuorman osittaisena keventämisenä, tarvittavan laitteiston käyttö- ja lainausmahdollisuutena sekä prototyypin komponenttihankintojen muodossa.

Erityisesti kiitän kaikkia työtovereitani, jotka ovat osoittaneet kiinnostustaan projektiani kohtaan sekä jaksaneet kuunnella ylitsevuotavia selostuksiani projektista. Heiltä olen saanut positiivista palautetta ja intoa jatkaa kehitystyötä silloinkin kun työ on tuntunut toivottomalta.

Kaikkein lämpimimmät kiitokset osoitan vaimolleni Heli Peltoniemelle, joka soi minulle runsaasti vapaahetkiä sekä lepoa diplomityötäni varten esikoisemme työntäyteisen ensimmäisen vuoden lomassa. Lisäksi kiitän esikoistani Toukoa riemullisista yhdessäolon hetkistä työpäivien jälkeen.

Turussa 15.8.2006

Jussi Sirkiä

Sisällysluettelo

1	Työn tausta ja tavoitteet	7
2	Laitteen määrittely lähtötiedoista	9
2.1	Ohjattavat moottorit	9
2.2	Harjattoman tasavirtamoottorin ohjaaminen	11
2.2.1	Pulssisuhteen ohjaus	12
2.2.2	Kommutoinnin ajoitus	13
2.2.3	Käynnistys	14
2.3	Käyttäjäraja-alue	17
2.4	Yleisiä vaatimuksia ohjaimelle	18
3	Tehoasteen suunnittelu	20
3.1	Kanavatransistorien valinta ja mitoitus	21
3.1.1	Kotelon valinta	21
3.1.2	Kanavan polariteetti	22
3.1.3	Transistorien jännitemitoitus	22
3.1.4	Transistorien häviöteho	23
3.2	Virta-anturi	24
3.2.1	Hall-ilmiöön perustuva virta-anturi	25
3.2.2	Virran resistiivinen mitta	26
3.3	Tulojännitteen suodatus	28
4	Ohjausosan suunnittelu	31
4.1	Prosessoriyksikkö	31
4.2	Kytkimien ohjaus	32
4.3	Jänniteregulointi	35
4.4	Vaihejännitteiden mitta	36
4.5	Piensignaalien käsittely	38
5	Ohjelmistokehitys	40
5.1	Laitteen alustus	40
5.2	Servopulssin mittaaminen ja käsittely	42
5.3	Virran mitta	44
5.4	Ohjauksen käynnistys	44
5.5	Kommutoinnin ajoitus	46
5.6	Moottorin poiskytkentä	49
6	Ohjaimen toiminta	50
6.1	Käynnistyminen	50
6.2	Matalan pyörimisnopeuden alue	51
6.3	Korkean pyörimisnopeuden alue	52
6.4	Täyden pulssisuhteen toiminta	54
6.5	Kommutoinnin ajoituksen tarkkuus	54
6.6	Ohjaimen maksiminopeus	55
6.7	Jatkokehitys	57
6.7.1	Laitteiston jatkokehitys	57
6.7.2	Ohjelmiston jatkokehitys	57
7	Johtopäätökset	59
	Lähteet	60
	Liitteet	
	LIITE 1: Prototyypin kytkentäkaavio	
	LIITE 2: Prototyypin ohjelmisto	

Lyhenneluettelo

BLDC	BrushLess Direct Current, harjaton tasavirtakäyttö
DC	Direct Current, tasavirta
FET	Field Effect Transistor, kanavatransistori
IGBT	Insulated Gate Bipolar Transistor, eristehilatransistori
LCD	Liquid Crystal Display, nestekidenäyttö
LRK	Moottorityyppi kehittäjiensä Ludvigin, Retzbachin ja Kühlfussin mukaan
PLL	Phase Locked Loop, vaihelukittu silmukka
PMSM	Permanent Magnet Synchronous Motor, kestopagneettitahtikone
PWM	Pulse Width Modulation, pulssisuhdemodulaatio
USB	Universal Serial Bus, tietokoneen laajennusväylä

1 Työn tausta ja tavoitteet

Harrasteilmailijoiden lennokeissa on vuosikymmenet suosittu voimanlähteenä kaksitahtista polttomoottoria, joka pääasiallisesti käy metanolin ja öljyn seoksella. Polttomoottorista saadaan suuri teho suhteessa moottorin ja sen välittömien oheiskomponenttien (polttoaine ja tankki) painoon nähden, mutta suuria ongelmia aiheuttavat voimakas, kimeä käyntiääni sekä voiteluaineen leviäminen lennokin ulkopinnoille. Öljyn leviämisen estämiseksi puuosat täytyy käsitellä rakennusvaiheessa öljyn pitäviksi, mikä puolestaan aiheuttaa lisää kustannuksia, painoa sekä ylimääräisiä työvaiheita lennokkia rakennettaessa, ja lisäksi koneen puhdistustarvetta irtoöljystä lentojen välissä ja lennätyksen jälkeen. Polttoaine on myös kohtalaisen kallista ja nykyisten tiukkojen vaarallisten aineiden kuljetussääntöjen vuoksi paikoin hankalaa hankkia. Tiheään asutetuilla seuduilla lentomelu saattaa aiheuttaa eripuraa asukkaiden ja harrastajien välille, ja monissa paikoissa onkin lennätyksellä aikarajoitus tai jopa polttomoottoreiden käyttökielto.

Yhdeksänkymmentäluvulta lähtien sähkömoottorit ovat saaneet yhä laajemman jalansijan lennokkien voimanlähteinä, mikä osin johtuu langattoman kulutuselektroniikan suuresta suosiosta. Entisen kokoisista ja painoisista akkupaketeista saadaan tätä nykyä moninkertainen kapasiteetti, ja myös sallitut purkuvirrat ovat kasvaneet. Näin sähköisten voimanlähteiden teho-painosuhte on kasvanut vertailukelpoiseksi polttomoottorien kanssa.

Alunperin lennokkien sähkömoottorit olivat tavanomaisia harjallisia tasavirtamoottoreita, joiden ohjaamiseen ei tarvittu sen kummempaa kuin yksi puolijohdekytkin katkomaan syöttöjännitettä. Moottorin nopeus seurasi hyvällä tarkkuudella kytkimen ohjauksen pulssisuhdetta, eikä pulssien taajuus ollut kovinkaan kriittinen: mikä tahansa taajuus parista sadasta hertsistä muutamaan kymmeneen tuhanteen oli käyttökelpoinen harrastajalle. Tee-se-itse-moottoriohjaimet ovatkin kuuluneet olennaisena osana sähkölennokkeihin, ja erilaisia kytkentöjä on voinut löytää alan lehdistä sekä harrastajien kotisivuilta ja foorumeista. Kytkentöjen toteutustavat ovat vaihdelleet yksinkertaisesta analogiaelektroniikasta monipuolisiin mikrokontrolleriohjaimiin.

Nykyinen suuntaus on varustaa sähkölennokit harjattomilla moottoreilla, joiden etuna on erinomainen hyötysuhde (jopa yli 95% akusta akselille) sekä lähes loputtoman pitkä käyttöikä, sillä ainoa kuluva osa ovat helposti vaihdettavat laakerit. Näiden moottoreiden huonona puolena kuitenkin on vahvojen kestopagneettien, moottorin ohjaimen tehokytkimien ja niiden ohjaukseen tarvittavien oheiskomponenttien sekä melko tehokkaan mikrokontrollerin tarve, mikä kasvattaa teholähteen hintaa. Valmiita kytkentöjä ei ole tähän mennessä montaa julkaistu, vaan suurin osa tällä hetkellä käytetyistä harjattomien moottoreiden ohjaimista on kaupallisia ratkaisuja, joiden hinta vaihtelee 60 eurosta jopa 400 euroon. Varsinainen moottorin ohjaamiseen tarvittava sähköinen kytkentä on hyvin tunnettu, mutta useimmiten kytkennän perusta on tehokas mikrokontrolleri, jonka ohjelmiston kehittämisessä harrastajilla on ollut suuria ongelmia. Viime aikoina muutamat innokkaat harrastajat ovat onnistuneet tekemään ohjelmistoja, jotka toimivat kuten kaupallisetkin, mutta näiden ongelmana ovat huonot hitaan käynnin ja eritoten käynnistyksen ominaisuudet käytettäessä erilaiset parametrit omaavia moottoreita samalla ohjelmistolla.

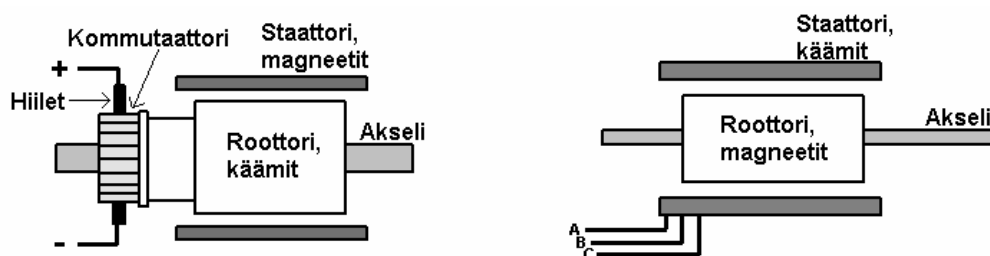
Tämän työn tavoitteena oli kehittää sekä laitteisto- että ohjelmistotasolla moottoriohjain, joka toimii suurillakin nopeuksilla, ja pystyy suorittamaan käynnistyksen riittävän luotettavasti moottorityypistä riippumatta ilman että käyttäjän tarvitsee muuttaa ohjaimen parametrejä moottoria vaihtaessaan. Loppujen lopuksi ohjaimen valmistusohjeet ja prosessorin lähdekoodi pyritään jatkokehittelyn jälkeen julkaisemaan Internetissä, jotta lennokkiharrastajien yhteisö voi saada täyden hyödyn työstä ja mahdollisesti kehittää ohjainta edelleen.

2 Laitteen määrittely lähtötiedoista

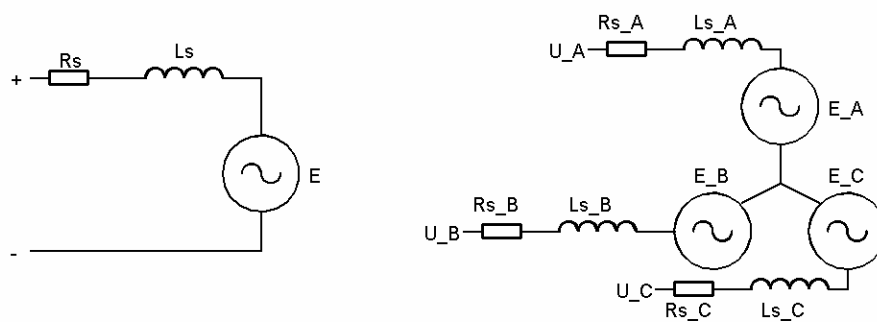
Tässä kappaleessa esitellään laitteen perusvaatimukset, eli selvitetään minkälainen on tyypillinen ohjattava moottori, ja miten sitä tulisi ohjata. Lisäksi käydään läpi laitteen tyypilliset käyttöolosuhteet ja moottorin kuorma sekä käyttäjärajapinta.

2.1 Ohjattavat moottorit

Harjallisen moottorin rakenne on melko monimutkainen harjojen ja kommutaattorin muodostaman tarkan kokonaisuuden takia. Harjattomien moottoreiden kokoonpano sen sijaan on varsin yksinkertainen, jopa siinä määrin, että viime aikoina harjattomien moottoreiden omakätinen rakentaminen on suorastaan räjähtänyt käsiin. Huomattavan monet alan liikkeet myyvät jo valmiita staattorilevyjä, kestopagneetteja, kuparilankaa ynnä muita tarpeita moottorinrakentajille. Paljon käytetään myös rikkoutuneista PC-laitteista, kuten kovalevyistä ja CD-asemista, talteen otettuja staattoreita, laakereita ja muita osia.



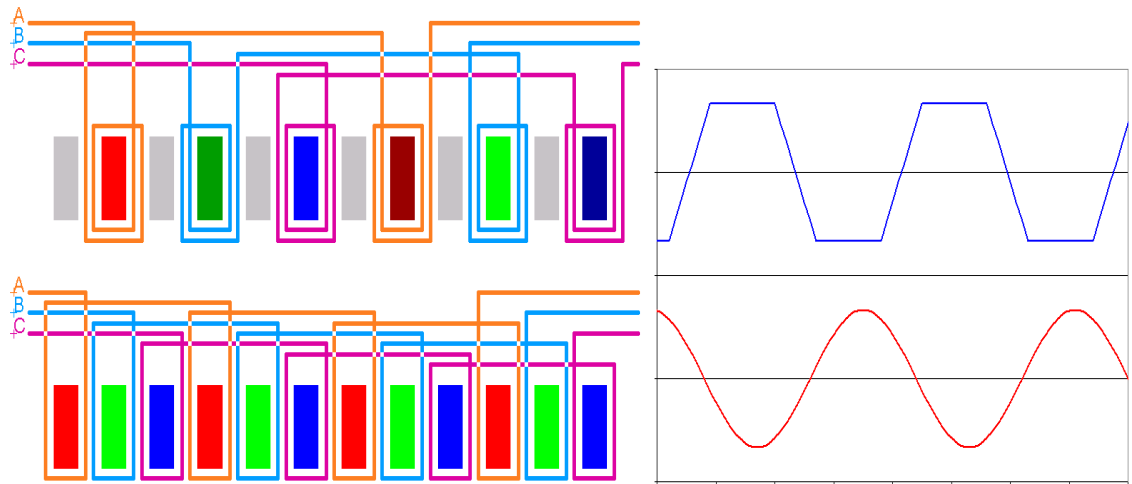
Kuva 1. Harjallisen ja harjattoman tasavirtamoottorin yksinkertaistetut poikkileikkaukset.



Kuva 2. Harjallisen ja harjattoman tasavirtamoottorin yksinkertaistetut sähköiset sijaiskytkennät.

Itse rakentamalla moottorin voi suunnitella tarkasti haluamaansa tarkoitukseen, oli käyttökohde sitten liidokin apumoottori, joka suuren potkurinsa kanssa vaatii suurta vääntömomenttia ja pieniä kierroksia, tai suihkukoneen pienoismallin puhallin, jonka pieni monilapainen potkuri saattaa pyöriä useita kymmeniä tuhansia kierroksia minuutissa. Moottorin kierrosnopeuteen tietyllä syöttöjännitteellä voi päämittojen lisäksi vaikuttaa pääasiassa kolmella tavalla: kääminnän kierroksia muuttamalla, moottorin napalukua vaihtamalla tai erilaisilla moottorin käämi-magneettiyhdistelmillä. Perinteisissä kestopagneettitahtikoneissa (PMSM, Permanent Magnet Synchronous Machine) jokaista kolmea käämiä kohti on yksi tai useampi napapari, ja napaparilukua kasvattamalla saadaan kierroslukua pienennettyä ja vääntömomenttia kasvatettua. Kierrosluvun merkittävä pienennys vaatii kuitenkin suuren määrän magneetteja ja käämejä, joten moottorista tulee työläs tehdä.

Nykyinen moottoreiden tee-se-itse-rakentaminen pohjautuu harjattomista tasavirtamoottoreista (BLDC, BrushLess Direct Current) Puolassa Wrocław'n teknillisessä korkeakoulussa tehtyyn tutkimukseen [1]. Saksalaiset harrastajat Christian Lucas, Ludvig Retzbach ja Emil Kuerfuss sovelsivat tutkimustulokset käytäntöön ja nimesivät moottorityypin sukunimiensä etukirjainten perusteella LRK-moottoriksi [2]. Retzbachin kirjoittama artikkeli aiheesta alan lehdessä johti muutamassa vuodessa kyseisen moottorityypin nykyiseen suureen suosioon, sillä LRK-tekniikalla moottorista saadaan parempi vääntömomentti samankokoiseen perinteiseen kolmivaihemoottoriin nähden, ja käämintäkin on helpompaa. Eritoten isoja potkureita pyöritettäessä päästään joissakin tapauksissa jopa kokonaan eroon alennusvaihteistosta moottorin ja kuorman välillä. Tämä parantaa merkittävästi koneen painoa, hyötysuhdetta, äänitasoa ja luotettavuutta. LRK-moottorin jännitemuoto on itsessään trapetsoidinen, joten se soveltuu erinomaisesti käytettäväksi harjattomana DC-moottorina, toisin kuin perinteiset moottorit, joiden aaltomuoto on pääasiassa siniä.



Kuva 3. LRK- (ylempi) ja perinteinen kolmivaihekäämintä sekä niille ominaiset aaltomuodot.

LRK-tekniikassa joka toinen käämihammas (harmaalla) jätetään käämimättä ja perättäiset yhden vaiheen käämit ovat 180 asteen vaihesiirrossa keskenään (vaalea ja tumma).

Tällä hetkellä itse rakennetut moottorit jakautuvat karkeasti kahteen ryhmään: isommassa koossa (staattorin halkaisija yli 30mm) suositaan pääasiassa LRK-periaatetta, kun taas pienemmissä, ns. CD-ROM-moottoreissa (staattorin alkuperäisen käytön mukaan) on yleensä käytössä perinteinen kolmivaihekäämintä. Tämä johtuu pääasiassa LRK-tekniikan edellyttämästä staattorin hampaiden lukumäärästä (tulee olla jaollinen kuudella), kun CD-asemissa käytetään yleensä yhdeksänhampaisia staattorilevyjä. Tehdastekoisissa lennokkimoottoreissa käytetään vaihtelevasti kumpaakin tekniikkaa kokoluokasta riippumatta.

Tässä työssä suunniteltu laite on ensisijaisesti tarkoitettu LRK-tyyppisen tai vastaavan trapetsoidaalisen aaltomuodon moottorin ohjaukseen, mutta sitä tullaan käyttämään myös CD-ROM-moottoreiden kanssa. Tavoite oli saada aikaan luotettava käynnistys ja käynti eri moottoreilla ilman laitteen parametrien manuaalista muuttamista. Periaatteessa ohjaimeen olisi ollut mahdollista toteuttaa ohjelmisto-osio, joka käyttäjän aloitteesta tutkii moottorin tärkeimmät parametrit ja käyttää niitä hyväkseen ohjauksessa. Tällaista tekniikkaa käytetään jo mm. ABB:n teollisuustasajuusmuuttajissa.

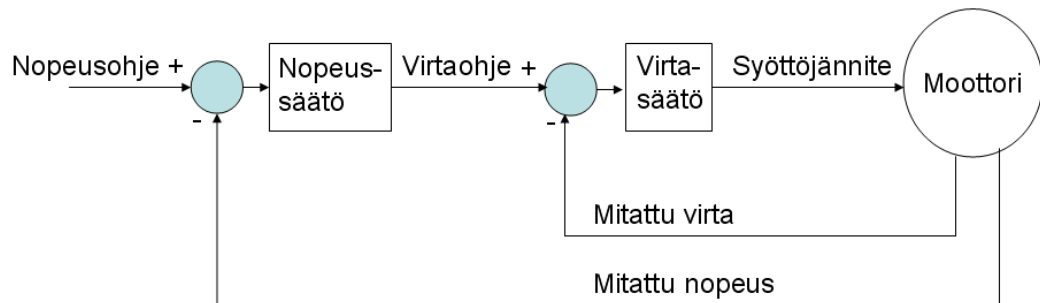
2.2 Harjattoman tasavirtamoottorin ohjaaminen

Normaalia DC-moottoria ohjataan tasajännitteellä, joka yleensä toteutetaan laskevalla katkojalla, jolloin lähtöjännitteen keskiarvo on katkojan tulojännite kertaa pulssisuhde.

Moottorin ominaisuudet määrittelevät millaisen virran tämä jännite kullakin pyörimisnopeudella aiheuttaa, ja mekaaniset hiiliharjat hoitavat sähköön vaiheistuksen roottorissa. Harjattomassa tasavirtamoottorissa jännite-virtasuhde pysyy koko lailla samana, mutta ohjaimen pitää huolehtia myös hiilien tehtävästä, eli vaiheiden kommutoinnista oikealla ajoituksella. Tämä monimutkaistaa elektroniikkaa huomattavasti.

2.2.1 Pulssisuhteen ohjaus

Teollisuudessa ja muissa tarkkaa ohjausta vaativissa sovelluksissa moottorin ohjaaminen perustuu sisäkkäisiin ohjaussilmukoihin. Sisin silmukka säätää moottorin virtaa, ja sitä kautta kulmakiihtyvyyttä, pulssisuhdetta muuttamalla. Virtasäädön ulkopuolinen silmukka ohjaa kiihtyvyyttä halutun nopeuden perusteella, ja mahdollinen uloin silmukka ohjaa nopeutta haluttavan liikkeen perusteella [3,4]. Moottorin ja kuorman ominaisuuksien perusteella kaikki säädöt voidaan mitoittaa optimaalisen nopeiksi ja tarkkoiksi.



Kuva 4. Moottorin ohjaus sisäkkäisten säätösilmukoiden avulla.

Lennoikkikäytössä monen sisäisen silmukan tarkka toiminta on liian monimutkainen toteutettavaksi kustannustehokkaasti sellaisenaan. Peruskäyttäjähalu haluaa ohjaimen, jonka voi liittää koneeseensa ja alkaa käyttää sitä. Pitkällisten säätöjen tekeminen ja moottorin ja kuorman parametrien mittailu tai arvailu on erittäin epätoivottavaa. Niinpä säätö toteutetaan usein pelkän sisimmän silmukan avoimena versiona, ts. käyttäjä säätää haluttua pulssisuhdetta, ohjain rajoittaa pulssisuhteen muutosnopeudet sopivalle tasolle. Moottorin virtaa ja nopeutta ei siis säädetä suoraan. Virta kyllä mitataan, mutta vain jotta ohjain selviäisi väärinkäyttöä vastaan. Moottori saa ohjaimen puolesta vaikka sulaa. Pyörimisnopeus on tietysti ohjaimella kommutointien hoitamiseksi tiedossa,

mutta varsinainen nopeuden takaisinkytkentä hoituu käyttäjän havaintojen ja toimintojen perusteella. Poikkeuksena ovat esimerkiksi radio-ohjattavat helikopterit, joissa pyritään roottorien vakiopyörimisnopeuteen. Näissä ohjaus voidaan toteuttaa varsinaisena kierrosnopeussäätönä tulkitsemalla käyttäjän ohjaus nopeusohjeeksi. Nopeusohjetta pyörimisnopeuteen vertaamalla saadaan pulssisuhteen ohjearvo, jonka perusteella ohjain voi toimia kuten suorassa pulssisuhteen ohjauksessakin.

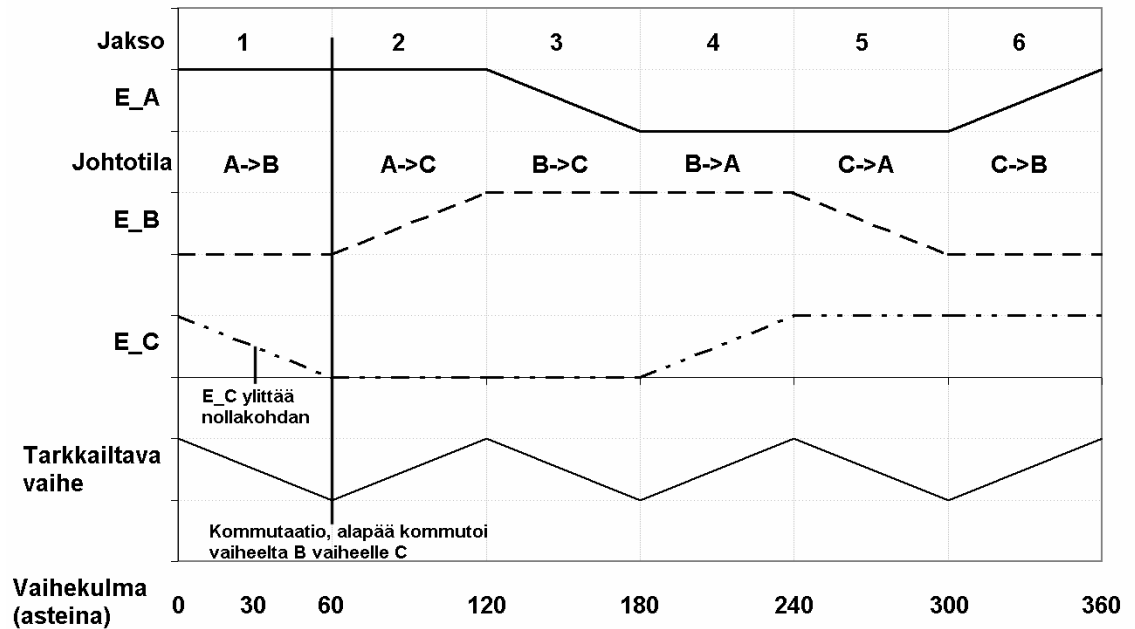
2.2.2 Kommutoinnin ajoitus

Harjattoman kolmivaiheisen DC-moottorin sähköinen kierros koostuu kuudesta jaksosta, joista jokaisessa kaksi vaihetta kolmesta on kytkettynä tulosähköön tasaisen pulssisuhteen PWM-modulaatiolla. PWM-modulaatiolla ei pyritä ohjaamaan vaihejännitettä sinimuotoiseksi kuten vaihtosähkökoneilla. Näin jokainen vaihe erikseen tarkasteltuna on hyvin lähellä perinteisen DC-moottorin ohjaamista H-sillalla, mistä johtuu moottorityypin nimikin.

Roottorin hetkellinen asento on helppo mitata magneettikenttää tutkivilla Hall-antureilla, joita kolme kappaletta sopivasti staattorin kehälle asettelemalla saadaan suora ohjaussignaali kullekin kuudelle kommutoinnille. Anturien pitää kuitenkin olla hyvin tarkasti oikeilla paikoilla, mistä aiheutuu turhaa monimutkaisuutta moottorin mekaniikkaan sekä tarve viidelle lisäjohdolle (kolme signaalia, maa ja anturien käyttöjännite) ohjaimelta moottorille. Niinpä lennökkikäytössä ei käytännössä katsoen käytetä enää lainkaan anturillisia ohjaimia, vaan kommutointitapahtumat perustuvat moottorin jännitemuotoihin.

Jokaisen jakson aikana kolmas vaihe on kytkemättömässä tilassa ja siihen indusoituu jännite moottorin liikkeessa. Tämä jännite on joko nouseva tai laskeva ramppi jaksosta ja pyörimissuunnasta riippuen, ja ideaalisesti jännite kohtaa ohjattujen vaiheiden jännite-eron keskikohdan jakson puolivälissä. Tätä keskikohtaa kutsutaan yleisesti nollakohdaksi, sillä kyseisen vaiheen indusoitunut jännite ylittää silloin nollatason. Kommutoinnista nollakohdan ylitykseen kuluva aika on yksi kahdestoistaosa eli 30 astetta yhdestä sähköisestä kierroksesta. Koska elektroniikan näkökulmasta mekaaninen kiihtyvyys on hyvin pieni, eli mekaaninen kulmanopeus on lähes vakio, kuluu nollakohdan ylityksestä seuraavaan kommutointiin normaalitilanteessa sama 30 astetta.

Niinpä moottorin ohjaus monissa nykyisissä ohjaimissa toteutetaan suoraan käyttäjän käskyjen mukaan pulssimodulaatiota säätämällä. Pulssinleveyden muutos aiheuttaa moottorin pyörimisnopeuden muutoksen, ja vapaan vaiheen tarkkailu antaa elektroniikalle takaisinkytkennän kommutointia varten.



Kuva 5. Moottorin indusoituneet jännitteet sekä kommutoinnit ja nollakohdan ylitykset yhden sähköisen kierroksen aikana, kun pulssileveysmodulointi on huipussaan (PWM on päällä koko ajan).

2.2.3 Käynnistys

Hankalin asia harjattoman DC-moottorin ohjaamisessa on yleisesti ottaen käynnistys, sillä takaisinkytkentä indusoituneesta vaihejännitteestä on riippuvainen kierrosnopeudesta. Nollanopeudella indusoitunutta jännitettä ei ole, ja alhaisilla nopeuksilla vapaan vaiheen jännite muuttuu jakson aikana vain vähän. Jännitteen tarkkailun pitäisi siis olla hyvin tarkka, mutta samalla kestää jännitemuodossa esiintyvät häiriöt sekä korkea yhteismuotoinen jännite. Käytännössä tämä on vaikea toteuttaa. Useiden ohjaimien alin toimintanopeus on melko korkealla, ja käynnistys onnistuu vain jos tähdätään heti alussa tätä rajaa korkeammalle. Koska kontrolleri ei tiedä moottorin sallittua maksimivirtaa, ei moottoria voida ajaa virtaohjattunakaan, jolloin saataisiin varma kiihtyvyys. Pikemminkin käytetään ”älykkäitä arvauksia” siitä, kuinka suuri

pulssisuhde käynnistyksessä moottorille syötetään, ja suoritetaan kommutoinnit ennalta määrätyllä taajuudella. Pulssisuhde ja taajuus joko riittävät antamaan kulloinkin vapaalle vaiheelle riittävän suuren indusoituneen jännitteen, jolloin siirrytään suljetun silmukan ohjaukseen, tai sitten ei. Tällöin keskeytetään ohjaus ja annetaan virheilmoitus käyttäjälle.

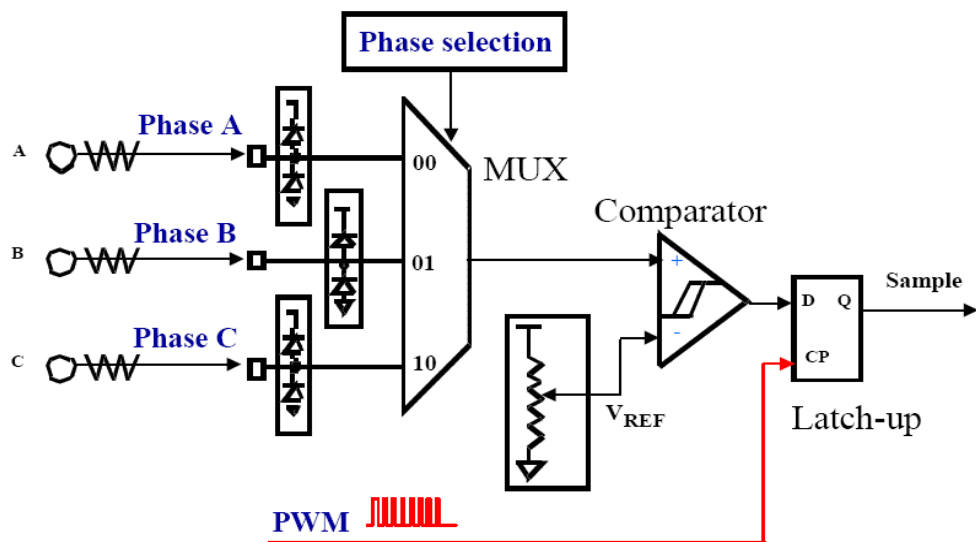
Lennokkikäytössä kuormana on aina potkuri tai puhallin, jonka oma massa on melko pieni. Siten käynnistyksessä tarvittava vääntömomentti ja virta-arvot ovat pieniä. Käynnistyksen pitäisikin siis onnistua luotettavasti siten, että moottorilla on minimipyörintänopeus, jota käytetään kun takaisinkytkentää ei vielä ole. Kun PWM:ää kasvatetaan, voimistuu takaisinkytkennän signaali ennen pitkää mitattavalle tasolle, jolloin siirrytään kommutoimaan takaisinkytkennän perusteella. Mitä tarkempi nollakohdan ylityksen mittaaminen pienillä kierroksilla on, sitä varmemmin ja nopeammin moottori käynnistyy.

S. Ogasawara ja H. Akagi ovat esittäneet [5], että käynnistyksessä ja matalilla kierroksilla voitaisiin tutkia indusoituneen jännitteen sijasta vaiheen virtaa. Heidän toteutuksessaan vain yhtä kytkintä kuudesta ohjataan kerrallaan PWM:llä, joten kun PWM on poissa päältä, kytketyn vaiheen normaalisti ei-johtavan kytkimen suojadiodi alkaa johtaa, tai lakkaa johtamasta, kun indusoitunut virta ylittää nollan. Tästä syntyy kytkimen ja suojadiodin yli diodin kynnysjännitteen suuruinen signaali. Diodin kynnysjännite on noin puoli voltia ja sillä on melko pieni riippuvuus virrasta, joten jo pienellä nopeudella ja siten pienellä indusoituneella virralla saadaan merkittävä jännitesignaali suojadiodin yli. Tekniikan tekee hankalaksi monimutkainen piiristö, jota tarvitaan vuorollaan jokaisen kytkimen suojadiodin johtamisen mittaamiseen.

Kuva 6. Ogasawaran ja Akagin mittausperiaate.

Jianwen Shao on kehittänyt diplomityönään [6] toisenlaisen matalan nopeuden tekniikan. Shao kytkee yläpäästä johtavan vaiheen PWM:llä vuoroin käyttöjännitteeseen ja vuoroin maahan. Vaihejännitettä tarkkaillaan vain PWM:n ollessa poissa päältä, jolloin vaiheeseen indusoituneen jännitteen nollakohdan ylitys on myös maapotentiaalin ylitys, sillä kaksi muuta vaihetta ja edelleen moottorin neutraalipiste ovat maissa, kun

PWM ei ole päällä. Koska ollaan kiinnostuneita vain nollan lähellä olevista signaalitasoista, voidaan muut tasot leikata pois yksinkertaisesti piirin sisäisillä suojadiodeilla maasta signaaliin ja signaalista piirin käyttö sähköön, ja vaiheen syöttämä virta rajoittaa sarjavastuksella ilman, että varsinainen signaali vaimentuu (kuva 7). Tekniikkaa on jatkokehitetty ST Microelectronicsin ST7MC-kontrolleriperheeseen [7], joka on nimenomaisesti suunnattu BLDC-ohjaukseen. Tässä kontrollerissa piirissä on lisäksi lähdöt, joilla signaalia voi myös vaimentaa halutulle tasolle. Vaimennettua vaihejännitettä voidaan tutkia suuremmilla nopeuksilla PWM:n ollessa päällä, sillä silloin indusoitunut jännite on huomattavasti suurempi, eikä vaimennus huononna tarkkuutta. Näin saadaan katettua moottorin koko toiminta-alue.



Kuva 7. Shaon mittausperiaate. Prosessori valitsee yhden kolmesta vaiheesta tarkkailtavaksi, komparaattori vertailee tasoa referenssitasoa vasten ja vertailun tulos näytteistetään PWM:n nousevalla reunalla.

Shaon ja ST Microelectronicsin mittaustapa on melko yksinkertainen toteuttaa, ja sitä voi edelleen yksinkertaistaa jättämällä kiinteätaajuisen näytteistykseen pois. Esimerkiksi PWM-taajuudella 20 kHz näytteistykseen aiheuttama viive nollakohdan tunnistuksessa voi pahimmillaan olla lähes 50 mikrosekuntia, mikä on jo huomattavan paljon. Jatkuvalla komparaattorin lähdön tarkkailulla saavutetaan pienempi viive, ja kun oikea signaali havaitaan, tarkistetaan että PWM oli oikeassa tilassaan mittaushetkellä.

2.3 Käyttäjäraja-alue

Käyttäjä ohjaa moottorinohjausta radiolaitteilla. Radiolta tuleva sähköinen viesti ohjaimelle on servosignaali, joka käyttää yksilinjasta pulssinleveysmoduloitua protokollaa. Eri radiolaitteiden valmistajien välillä on hyvin vähän eroja, pääasiassa suurimman kuuden valmistajan laitteet eroavat toisistaan vain signaalin polariteetin osalta. Aktiivinen signaali on useimmiten positiivinen. Radiovastaanottimelta kullekin ohjaimelle tulevat komennot toistuvat noin 20 millisekunnin välein, ja varsinainen komento on pulssi, jonka pituus vaihtelee yhden ja kahden millisekunnin välillä. Yksi millisekunti tarkoittaa ohjauksen toista ääripäätä ja kaksi millisekuntia toista, ja ohjaus tällä välillä on lineaarinen, eli puolitoista millisekuntia on ohjauksen keskikohta. Lähes kaikissa radiolaitteissa on vähintään suunnanvaihtokytkin, jolla saadaan vaihdettua antaako täysi kaasun siis yhden vai kahden millisekunnin pulssin. Lisäksi uudemmissa ns. tietokoneradioissa on myös mahdollisuus vaikuttaa ääripäiden etäisyyteen sekä säätötikun asennon ja lähdön väliseen funktioon, ts. ohjausvasteesta voidaan tehdä tahallaan epälineaarinen, tai maksimi- tai minimiohjausta voidaan rajoittaa.

Kunnollisen moottorinohjaimen perusvaatimus on, että kaasusäädön pitää olla minimissään ennen kuin ohjaus aktivoidaan. Näin vältetään tilanne, jossa käyttäjä on unohtanut kaasun päälle ja kytkee lennokkiinsa sähkö, jolloin ohjain antaa moottorille mahdollisesti voimakkaankin ohjauksen, ja käyttäjä voi loukkaantua tai laitteisto rikkoutua. Samoin mikäli radion signaali katoaa kokonaan, tai radio antaa pulsseja jotka ovat sallittujen rajojen ulkopuolella, tulisi säätimen sammuttaa moottori haavereiden minimoimiseksi.

Jo pelkkää tuotekehitystä, saati sitten loppukäyttöä varten on aina hyvä, että ohjaimessa on vähintään yksi ledi, jolla voidaan antaa palautetta kehittäjälle tai loppukäyttäjälle. Moottorikäytössä on myös mahdollista antaa palaute äänellä, eli syöttämällä sopivaa taajuutta moottorin joillekin vaiheille, jolloin käämitys ja roottori muodostavat heikkotehoisen kaiuttimen. Hienostuneempaa on tietysti käyttää laitetta kehitettäessä valmiita kehitysympäristöjä, joilla pääsee seuraamaan ja muokkaamaan reaaliaikaisesti ohjelman suoritusta. Tällöin käyttäjä saisi palautteen esimerkiksi LCD-näytön kautta. Näytölle ei kuitenkaan ole tämän kokoisessa laitteessa tilaa, ja sen sisällyttäminen laitteeseen nostaisi hintaa oleellisesti. Mikäli prosessorin nopeus ja ohjelman koko

suhteessa prosessorin muistin määrään sallivat, voidaan kehitysvaihetta ja käyttäjän säätämien asetusten vastaanottamista ja takaisinlukua varten ohjelmistoon lisätä kommunikointi PC:n kanssa tavallisen sarjaportin kautta.

Monipuolinen moottoriohjain antaa käyttäjälle mahdollisuuden muuttaa tiettyjä asetuksia laitetta käytettäessä, kuitenkin niin, että moottori on seisahtuneissa.

Normaaleja asetuksia ovat muun muassa nopeus- tai pulssisuhdeohjaus, tehon tai hyötysuhteen maksimointi, aktiivinen jarrutus tai vapaa pyöritys kun ohjaus on nollassa, ohjauksen nolla- ja maksimikohtien kalibrointi, kulmakiiktyvyyden säätö suoravedolle tai vaihteistolle jne. Tehdastekoisissa laitteissa käytetään erilaisia tapoja syöttää nämä tiedot ohjaimelle. Näistä yleisimmät ovat rivi pieniä kytkimiä, radion kaasutikku sekä PC:n USB- tai sarjaporttiliitäntä. Kytkimet vievät tilaa, kaasutikun käyttö taas vaatii jonkinlaisen palautteen nykyisestä säädöstä ja sen tilasta käyttäjälle ja saattaa monipuolisissa säädöissä olla hankala käyttää. Sarjaliikenne puolestaan vaatii kookkaat oheislaitteet, joita ei kentälle välttämättä voi ottaa mukaan, vaikka kannettava tietokone alkaa tosin olla yleinen näky jopa lennätuskentällä.

2.4 Yleisiä vaatimuksia ohjaimelle

Rakennettava ohjain suunniteltiin teholuokaltaan lennokkiohjaimien yläpäähän.

Maksimisyöttöjännitteeksi valittiin kolmekymmentä voltia, koska tätä isommat jännitteet vaativat niin paljon akkukennoja, että akkupaketin hinta muodostuu liian suureksi. Ohjaimen maksimiottovirta rajoitettiin testikäytössä toistaiseksi viiteen ampeeriin, mutta suunnitteluottotehona käytettiin jo alun pitäen noin puoltatoista hevosvoimaa eli 1200 wattia, mistä huippuottovirraksi saadaan noin 40 A.

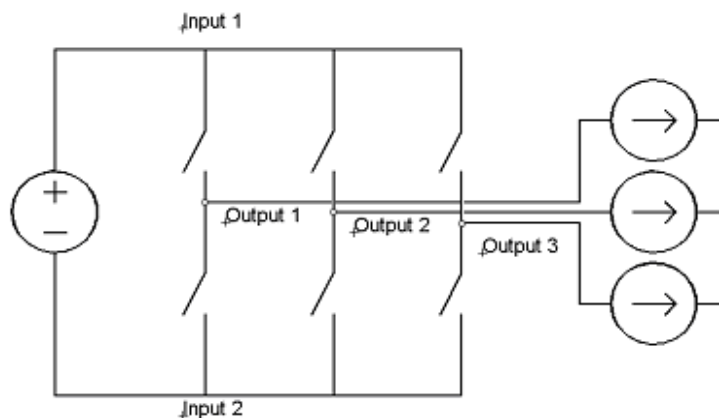
Pyörimisnopeus tämän kokoisilla moottoreilla on harvoin enemmän kuin 30 000 mekaanista kierrosta minuutissa, joten tämä asettaa vaatimuksen ohjaimen ohjelmiston suoritusnopeudelle. Tällä nopeudella pyörivät oikeastaan vain puhallinmoottorit, joiden napaluku on alhainen, joten voidaan arvioida, että 120 000 sähköistä kierrosta minuutissa riittäisi niille. Yleisimmin käytetty LRK-tekniikan variantti (6 käämiä, 14 magneettia) vaatii seitsemän sähköistä kierrosta jokaista mekaanista kierrosta kohti, joten 120 000 rpm sähköisenä riittää yli 17 000 rpm:n mekaaniseen nopeuteen, mikä taas riittää vallan hyvin, sillä LRK-moottorit on tarkoitettukin suurille potkureille pienillä nopeuksilla, tyypillisesti alle 10 000 rpm.

Laitteen pitää olla mahdollisimman kompakti, vaikka vielä tuotekehitysvaiheessa se ei välttämättä ole eduksi. Nelikerrospiirilevyn tai kahden pinotun kaksikerroslevyn käyttäminen on lähes pakollista, jotta pinta-ala saadaan minimoitua. Koska osa ohjaimesta kuljettaa suuria virtoja, olisi nämä kerrokset syytä erottaa signaalikerroksista ja käyttää niissä paksumpaa kuparin paksuutta piirilevyllä. Pikainen arviointi olemassaolevien ohjaimien pohjalta osoittaa, että koko laitteen pitäisi mahtua korkeintaan noin 50x40 mm kokoiselle levyille.

Lennokin nopeussäätimen häviötehon aiheuttama lämpenemä on teoriassa helppo siirtää ympäröivään ilmaan, kulkeehan lennokka jatkuvasti uuden viileän ilman halki. Käytännössä nopeudensäädin halutaan kuitenkin pitää aina kokonaan koneen sisällä, ja koneen rungon avaukset, joista jäähdytysilma pääsee sisään ja ulos, ovat hyvin pieniä. Niinpä säätimessä tulee olla kunnollinen jäähdytyslevy, joka siirtää lämmön tehokomponenteilta tehokkaasti ympäröivään ilmaan. Lisäksi säätimessä tulee olla lämpötilan tarkkailu, jotta moottorin tehoja voidaan vähentää laitteiston kannalta turvalliselle tasolle, mikäli jäähdytys osoittautuu puutteelliseksi.

3 Tehoasteen suunnittelu

Kolmivaiheisen moottorin ja tasajännitelähteen väliin tarvitaan 2×3 - eli kuusikytkiminen matriisi, jotta jokainen vaihe on täysin ohjattavissa tulosähkön alueella. Kytkennän resiprookkisuudesta johtuu, että matriisin toinen puoli edustaa jännitelähdettä ja toinen virtalähdettä, ts. virralla syötetään jännitettä tai päinvastoin. Kytkimien ohjauksella tulee varmistaa, ettei jännitelähde oikosulkeudu eivätkä virtalähteen navat avaudu. Käytännössä tämä on helpointa toteuttaa kuuden tehpuolijohdekytkimen muodostamalla kolmivaihesillalla sekä kytkimiin vastarinnankytketyillä diodeilla, jotka antavat moottorin induktiiviselle virralle kulkutien.



Kuva 8. Kytkinmatriisi ja lähteen ja kuorman sijaiskytkennät.

Puolijohdekytkimien tulee pystyä kytkemään nopeasti suuria virtoja, mutta jännitekestoisuuden tarve normaalissa lennökkikäytössä jää melko pieneksi: tyypillisesti alle 40 voltin. IGBT eli eristehilatransistori on kytkimenä liian hidas ja yleensä tarkoitettu suuremmille jännitteille, kun taas bipolaaritransistorit ovat hitaahkoja ja vaativat suuren kantavirran vuoksi kehittyneen kantaohjauspiiristön. Niinpä laitteessa päätettiin käyttää kanavatransistoreja (FET, Field Effect Transistor), joiden ohjaus on helppoa, toiminta nopeaa ja päästötilan häviöt hyvin pienet. Häviöt ovat pienentyneet harppauksin sen jälkeen kun autoteollisuus aloitti siirtymän vanhanaikaisesta reletekniikasta tehokkaampaan elektroniikkaan. Lisäksi kanavatransistoreissa on käytetystä tekniikasta johtuen sisäänrakennettuna vastarinnankytketty diodi, joten erillisiä diodeja ei tarvitse enää lisätä kytkimien rinnalle. Tämä säästää piirilevyllä

käytettävää pinta-alaa sekä painoa ja hintaa. Toisaalta sisäänrakennetun diodin ominaisuudet eivät pääse lähellekään erillisten diodien vastaavia, mistä aiheutuu merkittävä ohjaimen hyötysuhteen aleneminen. Diodien häviöt voi kuitenkin kiertää ohjaamalla kulloinkin aktiivisena olevia vaiheita siten, että diodit eivät ala johtaa, vaan virta pääsee kulkemaan avoimen kytkimen kautta, ts. kun yhden vaiheen toinen kytkin katkaisee virran, kytketään välittömästi saman vaiheen toinen kytkin johtamaan. Tätä synkronisessa tasasuuntauksessa käytettävää tekniikkaa päätettiin käyttää suunniteltavassa laitteessa ja tyytyä pelkkiin kanavatransistoreihin.

3.1 Kanavatransistorien valinta ja mitoitus

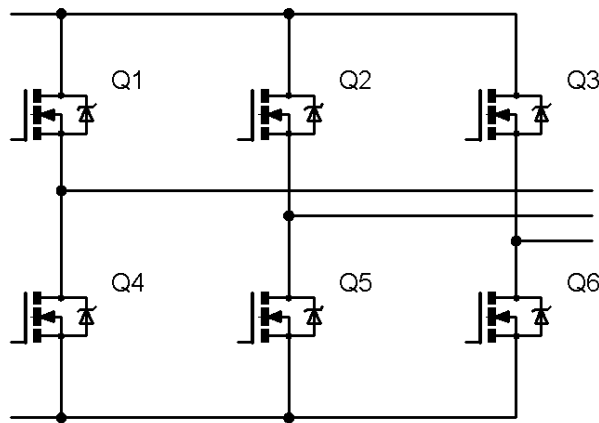
Kanavatransistorien valintaan vaikuttavat lähinnä neljä pääseikkaa: haluttu kotelon koko, jännitekestoisuus, päästötilan resistanssi sekä hilan kapasitanssi.

3.1.1 Kotelon valinta

Perinteisesti tehokytkimet ovat olleet TO-220- tai vielä isommissa läpiladottavissa koteloissa, jotka on helppo kiinnittää jäähdytyslevyyn. Kotelo on kuitenkin erittäin suuri lennökkikäyttöön, ja pelkästään komponentin jalkojen pituus aiheuttaa suurehkon induktanssin ohjaimien ja varsinaisen kytkimen välille. TO-220-paketista johdettu TO-263 eli D2-PAK on vastaavan kokoinen, mutta pintaliitosversio aiheesta. Kotelon leveys on 10,7 mm ja pituus 15,4 mm, korkeus piirilevyn pinnasta on noin 4,5 mm. Standardiksi useimmissa tehdasvalmisteisissa lennökkien harjattomissa ohjaimissa on kuitenkin vakiintunut SO-8-kotelo, joka on 5 mm leveä ja pitkä sekä vain yhden millimetrin korkuinen. Näitä on saatavissa useilta valmistajilta erittäin hyvillä ominaisuuksilla - jopa niin hyvillä, että yleensä jäähdytyksestä vastaa vain koteloiden päälle asennettu sileä alumiinilevy. SO-8 on pintaliitoskotelo, joten se kuluttaa piirilevytilaa vain levyn yhdeltä puolelta. Niinpä laitteeseen päätettiin valita kytkimet SO-8-kotelossa. Markkinoilla on myös paljon koteloita, joiden koko ei ole paljon valittua suurempi, mutta lämmönjohto-ominaisuuksia on parannettu esimerkiksi toteuttamalla kotelon ylä- tai alapinta keraamisena tai metallisena. Nämä kotelot ovat kuitenkin hyvin usein valmistajakohtaisia, joten korvaavia tyyppjeä on saatavuusongelmien sattuessa vaikea löytää.

3.1.2 Kanavan polariteetti

Kytkinmatriisiin ylä- ja alapään kytkimet voisi periaatteessa toteuttaa komplementaarisilla kytkimillä, eli sijoittamalla N-kanavainen alas ja P-kanavainen ylös. Tätä tekniikkaa käytetään joissakin alhaisen jännitteen ohjaimissa, koska kytkimiä voidaan tällöin ohjata lähes suoraan logiikalla ilman kalliita ja tilaavieviä ajuripiirejä. Tyypillisesti N-kanavaiset fetit ovat kuitenkin nopeampia ja vähempihäviöisiä kuin P-tyypin vastapuoliskonsa. Lisäksi tehtävän ohjaimen tulee joka tapauksessa kestää korkeita jännitteitä, joten hilaohjauksesta ei saa logiikkatasoista missään tapauksessa. Niinpä on parempi optimoida kytkimien häviöt kuin niiden ohjauspiirit. Valitaan siis käytettäväksi vain N-kanavaisia transistoreja.



Kuva 9. N-kanavaisilla FETeillä ja niiden ominaisdiodeilla toteutettu kytkinmatriisi.

3.1.3 Transistorien jännitemitoitus

Kytkimien tulee kestää tulojännitteen vaihtelualue, joten tässä laitteessa kytkiminä olisi hyvä käyttää vähintään 40 voltin kytkimiä. Mitä isomman jännitteen kytkin kestää, sitä suuremmat ovat sen päästötilan resistiiviset häviöt tietyllä virralla. Liian suurta jännitekestoa ei siis kannata valita, mutta selvä marginaali fetillä pitää silti olla. Autoteollisuus on siirtymässä 42 voltin järjestelmiin, joissa käytetään vähintään 60 voltin kanavatransistoreja. Moni valmistaja tarjoaa jo nyt useita tällaisia kytkimiä erilaisilla spesifikaatioilla. Pikainen haku erään suuren kytkinvalmistajan verkkosivuilla [8] näyttää useita SO-8-koteloisia kytkimiä, joista voidaan valita sopiva toteutettavaan moottoriohjaimeen.

Taulukko 3.1. Fairchild Semiconductorsin 40:n ja 60:n voltin SO-8-koteloiset FETit. [7]

Part Number	Polarity	Configuration	Package	Vds	Id	Rds (on)	R _{DS(on)} Max (Ω) @ V _{GS} =		Qg
							10V	4.5V	
FDS4470	N	Single	SO-8	40 V	12.5 A	9 mOhm	0.009 Ohm		45 nC
FDS4480	N	Single	SO-8	40 V	10.8 A	12 mOhm	0.012 Ohm		29 nC
FDS4672A	N	Single	SO-8	40 V	11.0 A	13 mOhm		0.013 Ohm	35 nC
FDS4770	N	Single	SO-8	40 V	13.2 A	7 mOhm	0.0075 Ohm		47 nC
FDS4780	N	Single	SO-8	40 V	10.8 A	10 mOhm	0.0105 Ohm		30 nC
FDS5670	N	Single	SO-8	60 V	10 A	14 mOhm	0.014 Ohm		49 nC
FDS5680	N	Single	SO-8	60 V	8 A	20 mOhm	0.02 Ohm		30 nC
FDS5690	N	Single	SO-8	60 V	7 A	28 mOhm	0.028 Ohm		23 nC
RF1K49154	N	Single	SO-8	60 V	2 A	130 mOhm	0.13 Ohm		14 nC

Haun tuloksena löydettyistä kytkimistä (taulukko 1) nähdään, että 40 V kestolla minimivastus päästötilassa on 7 milliohmia (FDS4770), mutta 60 V kesto nostaa minimivastuksen jo 14 mΩ:iin (FDS5670) eli kaksinkertaiseksi. Kummallakin transistorilla hilan vaatimukset ovat lähes samat; noin 2,9 nanofaradin tulokapasitanssi tai 50-70 nanocoulombin hilavaraus, joten kytkentähäviöt ovat melko yhteneväiset. Kytkintyyppiä vaihtamalla ohjaimesta saa siis helposti varioitua isovirtaisen, mutta alhaisemman jännitteen, ja toisaalta korkean jännitteen, mutta pienemmän virran version.

3.1.4 Transistorien häviöteho

Ensimmäiseen protoon valittiin 40 voltin FDS4470, jolla johtohäviöiksi muodostuu

$$P_{on} = I^2 R = (40A)^2 \cdot 0,009\Omega = 14,4W \quad (1)$$

eli varsin huomattava teho. Kolme kytkintä rinnakkain pudottaa kokonaishäviön kolmannekseen, ja kytkinkohtaisen häviön yhdeksännesosaan eli 1,6 W:iin, ja koska kukin vaihe johtaa maksimissaan kolmasosan ajasta, saadaan lopulta kytkinhäviöksi 533 mW. Tämän tehon normaali SO-8-kotelo kestää riittävällä jäähdytyksellä varsin hyvin.

Kytkimien johtohäviöitä voidaan vähentää laittamalla kytkimiä rinnan, mutta samalla hilan kapasitanssi kasvaa, mikä kasvattaa hilaohjauksen virrantarvetta sekä kytkentähäviöitä. Jos kantoja ohjataan 12 voltin jännitteellä ja kytkimen hilan varaus on 45 nC, on 20 kHz:in PWM-taajuudella yhden hilan ohjaukseen kuluva teho

$$P_{drive} = V_{GS} \cdot Q_G \cdot f = 12V \cdot 45nC \cdot 20kHz = 11mW \quad [9] \quad (2)$$

mikä on minimaalista johtohäviöihin nähden.

Varsinainen kytkentähäviö muodostuu ajasta, jonka moottorin virta kommutoi kytkimeltä diodille (tai toisinpäin), jolloin kytkimen yli vaikuttaa jännite ja virtaa kulkee sen läpi. Kommutoinnin keskellä virta on puolivälissä samoin kuin jännite, joten tällöin häviöteho on suurimmillaan. FDS4770:lle ilmoitetaan virran tyypilliseksi nousuajaksi 12 ns ja laskuajaksi 29 ns. Jos oletetaan että keskimääräinen kommutoinnin aikana syntynyt häviöteho on 70 % huipputehosta, ovat yhden vaiheen fettien kytkentähäviöt

$$P_{switch} = t \cdot f \cdot 0,70 \cdot \frac{I}{2} \cdot \frac{U}{2} = (12ns + 29ns) \cdot 20kHz \cdot 0,70 \cdot \frac{40A}{2} \cdot \frac{30V}{2} = 172mW \quad (3)$$

eli edelleen merkityksettömän pienet (kymmenesosa) johtohäviöihin nähden.

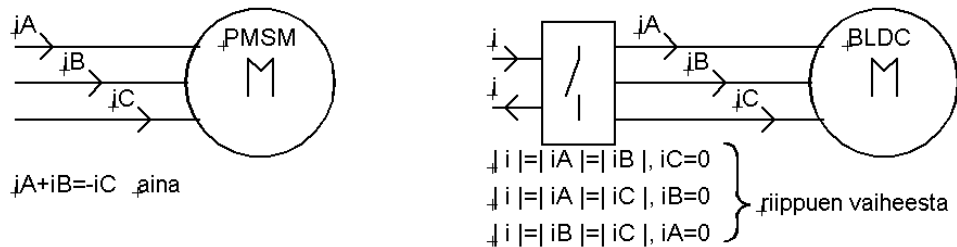
Tyydytään siis kolmeen FDS4470-piiriin per kytkinyksikkö.

3.2 Virta-anturi

Moottori ja sitä syöttävä akku pitää suojata ylivirtatilanteilta, joten vaiheiden virta pitää pystyä mittaamaan. Virta kulkee myös melko suorassa suhteessa moottorin vääntömomenttiin, mutta koska tässä työssä kehitetään yleiskäyttöistä ohjainta, joka ei välttämättä tiedä ohjaamansa moottorin ominaisuuksia, ei säätönäkökulma ole virran mittauksessa oleellinen.

Harjattoman DC-moottorin tilanteessa ei ole tarpeen mitata kaikkia vaiheita erikseen. Koska kerrallaan vain kaksi vaihetta johtaa sarjassa, vastaa tilanne tavallista laskevaa hakkurijännitelähdettä, jolloin jännitelähteestä otettavan virran keskiarvo on aina lähes sama kuin moottorin johtavien vaiheiden virran keskiarvo kertaa ohjauksen pulssisuhde. Hajainduktanssi vapaassa vaiheessa sekä ääretöntä pienempi induktanssi ohjatuissa vaiheissa aiheuttavat pienen virheen aktiivisten vaiheiden virran ja ottovirran

suhteeseen: Pieni osa ottovirrasta jatkaa kulkuaan vapaan vaiheen kautta. Pienuudesta johtuen tämä voidaan kuitenkin jättää huomiotta. Virran mittaukseen riittää siis yksi sopivasti mitoitettu anturi kytkinmatriisiin ja syöttävän lähteen toisen kiskon välille.



Kuva 10. Moottorin vaiheiden virrat AC-moottorilla ja harjattomalla tasavirtamoottorilla.

Tasavirtamoottorin ohjaukseen riittää syöttövirran mittaus, sillä yhden vaiheen virta on aina lähes nolla ja kahden muun vaiheen virran itseisarvo on yhtä kuin syöttövirta.

Kolme pääanturityyppiä virran mittaukseen ovat resistiivinen, Hall-ilmiöön perustuva ja induktiivinen anturi. Näistä vain kahdesta ensimmäisestä saadaan selville myös virran DC-taso, mikä tässä työssä on ehdoton vaatimus. Induktiiviset anturit ovat usein myös melko kookkaita, joten niitä ei tässä käsitellä sen enempää.

3.2.1 Hall-ilmiöön perustuva virta-anturi

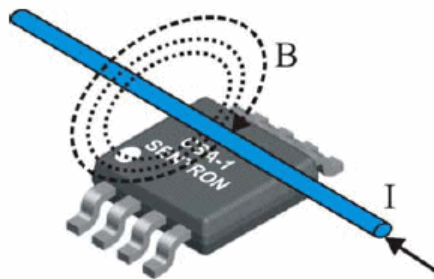
Hall-ilmiössä ulkoinen magneettikenttä aiheuttaa virtaa kuljettavaan ohueen johtimeen jännite-eron poikittaisesti virtaan ja magneettikenttään nähden. Tämä jännite on suoraan verrannollinen sekä virtaan että magneettikenttään, joten jos jompikumpi tiedetään, jännitteestä voidaan laskea suoraan toinen. IC-piirien valmistustekniikan kannalta on kätevämpää synnyttää vakiovirta johtimeen ja mitata magneettikenttää, mutta tällöin mitattava virta pitää ensin saada muunnettua jollakin suhteella magneettikentäksi. Virta-anturi koostuu yleensä toroidin muotoisesta suuren permeabiliteetin omaavasta materiaalista valmistetusta ”kerääjästä”, jonka aukosta mitattavan virran johdin kulkee. Virtajohtimen ympärilleen aiheuttama magneettikenttä [10] kulkee pääosin tässä toroidissa, ja kun toroidi on jostakin kohtaa kehänsä katkaistu, voidaan Hall-anturi asettaa syntyneeseen rakoon mittaamaan magneettikentän kautta alkuperäistä virtaa.

Ongelmana tällaisten anturien käytössä on toroidimateriaalin mekaaninen hauraus, piirilevyille asettamisen vaikeus sekä yleensä liian suuri koko miniatyyrilaitteisiin.

Aivan hiljattain on markkinoille kuitenkin ilmestynyt mikropiirejä, jotka joko pitävät koko mittauslaitteen sisällään [11], jolloin myös mitattava virta kulkee piirin läpi, tai sisältävät magneettikentän fokusointiin tarvittavat osat ja signaalinkäsittelyn [12], jolloin virta pitää ulkoisesti kuljettaa piirin läheltä esim. asettamalla komponentti piirilevyllä mitattavan vedon päälle. Jälkimmäisestä esimerkkejä ovat Sentron-yhtiön CSA-1-sarjan komponentit, jotka ovat kätevässä SO-8-kotelossa ja joiden maksimivirtaa rajoittaa vain se, kuinka kauas mitattavan johtimen voi asettaa (etäämpänä oleva johdin kasvattaa häiriöherkkyyttä piirin mitatessa kaikkien lähellä kulkevien virtojen summan). Tällaisessa ratkaisussa erinomaisena puolena on myös se, että signaalinkäsittely on täysin galvaanisesti erotettu mitattavasta johdosta, jolloin mittaus voidaan helposti suorittaa missä tahansa potentiaalissa olevasta johtimesta. CSA-1V:n hinta tuhannen kappaleen erissä on noin 4 euroa, mikä on varsin maltillinen hinta piirin ominaisuuksiin nähden.

Current Measurement A

The current conductor is situated above the chip.



A current conductor is placed at a certain distance above or beyond the chip. The magnetic field around a long current wire is described by $H = I / 2\pi r$. The flux density at the place of the sensor depends upon the current in the wire I and the distance between sensor and wire r :

$$B = \mu_0 * I / 2 \pi r.$$

Current Measurement B

The current conductor is situated below the chip (on pcb).



The CSA-1 current sensor is placed on a PCB directly over the current track. The current track can consist of one or several loops, depending on the maximum current to be measured. The sensor measures the magnetic field emanating from the current flowing through the tracks.

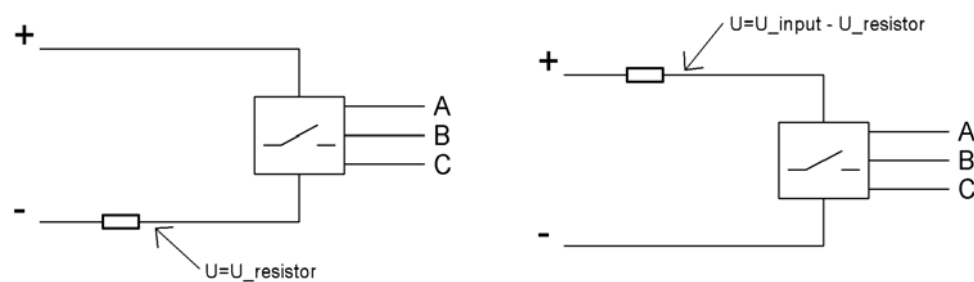
Kuva 11. Sentronin magneettikenttään perustuva virta-anturi CSA-1. [11]

3.2.2 Virran resistiivinen mittaus

Virta aiheuttaa aina resistanssin yli jännitteen, joten vastusta voi käyttää hyvin yksinkertaisena virta-jännite-muuntimena. Ongelmana on lähinnä vastuksessa aiheutuva

tehohäviö, joka kasvaa suhteessa virran toiseen potenssiin. Niinpä resistiivistä mittausta käytetään normaalisti pienielektroniikassa vain mitattaessa pieniä (alle 10A) virtoja. Tehohäviön minimointi vastusta pienentämällä aiheuttaa virta-jännitevahvistuksen pienenemistä, joten signaalia täytyy jatkokäsittelyä varten vahvistaa esimerkiksi operaatiovahvistinkytkennällä.

Signaalinkäsittelyn helpottamiseksi resistiivisessä mittauksessa käytetään usein negatiivista (maa-)kiskoa, jolloin virtasignaali näkyy suoraan vastuksen yli positiivisena signaalina maata vasten, eikä pienestä erosignaalista tarvitse poistaa suurta yhteismuotoista jännitettä. Tämä oli alkuperäinen ajatus tässä työssä, mutta vahvistinta valittaessa löytyi Texas Instrumentsilta INA195-piiri, joka on tarkoitettu nimenomaan virran mittaukseen positiivisesta kiskosta mahdollisimman vähillä oheiskomponenteilla. Niinpä se valittiin käytettäväksi prototyypissä. 40 ampeerin virta edellyttää erittäin pieni-impedanssista mittavastusta, jolle joka tapauksessa saattaa jäädä melko suuri häviöteho. Monissa teholähteissä käytetään metallilangasta tehtyjä kookkaitakin läpijuotettavia vastuksia, joilla ei tehonkestossa kuitenkaan päästä kuin kahteen wattiin [13]. Tämä ei ole sen parempi kuin 2512-kokoisella pintaliitosvastuksella [14]. Mitä pienempi vastus sitä pienempi hajainduktanssi sillä on, mikä tarkentaa mittausta.



Kuva 12. Virran resistiivinen mittaaminen negatiivisesta (vasemmalla) ja positiivisesta kiskosta.

Isojen koteloiden mukana vastukseen tulevat usein ns. Kelvin-kontaktit, jolloin vastuksessa on neljä kontaktia, kaksi tehonsyötölle ja kaksi jännite-eron mittaukselle. Tämä tekniikka antaa hyvän tarkkuuden suurilla virroilla, joilla piirilevyn vedotkin aiheuttavat suhteellisen suuren eron mittaustulokseen sen mukaan, miten johtimet on piirilevyllä sijoitettu. Viisi milliohmia on minimivastusarvo 2512-kotelossa, joka on kohtuuhintaan ja helposti saatavilla usealta valmistajalta. Yleisenä analogia- ja logiikkajännitteenä aiottiin käyttää viittä voltia, joten virranmittaussignaali on hyvä

skaalata valmiiksi tälle alueelle. INA195-piirillä vahvistus on sata, joten virranmittausvastuksella pitää näkyä 50mV signaali, jotta piirin lähdössä on viisi volttia. 40 ampeerin virralla voidaan laskea tarvittava vastusarvo ja kokonaishäviöteho:

$$R = \frac{U}{I} = \frac{50\text{mV}}{40\text{A}} = 1,25\text{m}\Omega \quad (4)$$

$$P = U \cdot I = 50\text{mV} \cdot 40\text{A} = 2\text{W} \quad (5)$$

Neljä viiden milliohmin vastusta rinnakkain antaisi siis halutun jänniteskaalauksen ja myös tehonkeston. Täytyy kuitenkin ottaa huomioon, että mikäli vastukset sijoitetaan aivan toisiinsa kiinni, lämmittävät ne toisiaan ja saattavat aiheuttaa ylikuumenemisongelmia etenkin keskimmäisiin vastuksiin.

Testausta varten virranmittausvastukseksi valittiin vain yksi kymmenen milliohmin vastus, jolla saadaan viiden ampeerin maksimitestivirta muutettua viiteen volttiin.

Vastukset 2512-kotelossa watin tehonkestolla ja viiden milliohmin vastuksella maksavat noin 90 senttiä kappaleelta ja INA195 saman verran, joten virtamittaus resistiivisesti toteutettuna tulee maksamaan lähes yhtä paljon (~4,5€) kuin edellä tutkitulla Sentronin Hall-anturillakin. Tilaa kuluu roimasti enemmän ja tehohäviöitäkin syntyy enemmän. Toisaalta laitteen tuotannon kannalta vahvistimelle ja vastuksille on mahdollista löytää useita korvaavia valmistajia, mikä on hyvä, kun taas erikoisen Hall-anturin osalta oltaisiin yhden ainoan toimittajan varassa. Jatkossa, kun Hall-virta-anturit varmasti yleistyvät, voidaan harkita sellaiseen siirtymistä.

3.3 Tulojännitteen suodatus

Viimeisenä teho-osion komponenttina piti kortin jännitesyöttöön kytkeä mahdollisimman iso kondensaattori, jolla tulon jännitepiikkejä rajoitetaan. Moottori toimii kuten virtalähde, joten sen oma virta on pitkälti tasaista, mutta ohjaimen tulovirta heilahtelee PWM:n mukaan kymmeniä ampeereja muutamissa mikrosekunneissa. Jäykälle jännitelähteelle, kuten akulle, tämä ei olisi mikään ongelma, mutta lopullisessa käyttötilanteessa voi laitteiden sijoittelun vuoksi olla pakko kytkeä akku melko pitkällä kaapeleilla ohjaimeen, ja kaapelin induktanssi yhdistettynä PWM-ohjaukseen aiheuttaa tuloon kytkentänopeuteen verrannollisia ylijännitepiikkejä. Lisäksi virran suuret vaihtelut johdoissa johtavat helposti radiotaajuisiin häiriöihin, jotka pahimmillaan

voivat haitata laitteen ohjaamista. Tuloon kytkettävän kondensaattorin tulee olla niin iso kuin käytännössä voidaan sallia ilman että laitteen koko kasvaa liikaa.

Kondensaattorin sarjaresistanssin taas tulisi olla mahdollisimman pieni, koska jännitepiikkien suuruus ja kondensaattorin omat häviöt ovat verrannollisia siihen.

Pintaliitoskondensaattorit kuluttavat paljon piirilevytilaa ja lisäävät tuotteen korkeutta merkittävästi. Parempi on käyttää joko aksiaalikondensaattoria piirilevyn päädyn suuntaisena tai useampaa radiaalikondensaattoria kortin reunan ulkopuolella.

Molemmissa tapauksissa piirilevytilaa kuluttavat pelkät juotoskohdat, kondensaattorin runko jää hivenen kauemmas kuumana käyvistä puolijohteista ja kondensaattori on mahdollista saada aivan kiinni tulojohtoihin.

Aksiaalikondensaattorit ovat nykyään huomattavasti vähemmän käytettyjä kuin radiaaliset johtuen lähinnä jälkimmäisten helpommasta piirilevyasennuksesta, mutta esimerkiksi Epcosilta [15] löytyy vielä autoteollisuudelle tarkoitettu sarja, jossa sallitut vaihtovirrat ovat moninkertaisia radiaalikondensaattoreiden vastaaviin nähden.

Esimerkiksi 470 μF :n, 63 V:n kondensaattorille luvataan 4,6 A virrankesto 85 asteen lämpötilassa ja 10 kHz taajuudella. Vertailun vuoksi Nichiconin PJ-sarjan 470 μF 63 V radiaalikondensaattori kestää vain 1,78 A 105 asteen lämmössä ja samalla taajuudella [16].

Valitaan huonommista ominaisuuksista huolimatta tarkastelun kohteeksi monin verroin yleisempi Nichiconin PJ-sarjan 50V 470 μF kondensaattori 12,5x25 mm (halkaisija x korkeus) kotelossa. Näitä saadaan tulevan kortin reunaan mahtumaan kaksi, joten yhteiseksi kapasitanssiksi muodostuu 940 μF ja sallituksi sykkeisyysvirraksi 20 kHz taajuudella 3A. Sarjavastus tällä kokoonpanolla on yhteensä noin 60 milliohmia.

Kondensaattorin vaihtovirran ja tulojännitteen sykkeisyyden laskeminen on monimutkainen toimitus, joten päätettiin luottaa kondensaattorin riittävyyteen sillä perusteella, että kaupallisissa laitteissa ei yleensä näe edes tämän kokoisia kondensaattoreita. Lisäksi prototyypistä voidaan haluttaessa mitata kondensaattorin vaihtovirta ja tulojännitteen vaihtelu.

Suuren kapasitanssiarvon kondensaattorin lisäksi on erittäin tärkeää, että myös suuritaajuisien virtojen kulkuteistä huolehditaan. Sekä syöttöjohtojen että

kytkinmatriisiin väliin pitää laittaa noin 100nF:n keraamiset kondensaattorit, jotka poistavat suurelta osalta suuritaajuiset häiriöt heti kytkentäsolmupisteen vierestä. Näin saadaan minimoitua radiotaajuiset häiriöt jo syntypaikallaan, sillä kytkentähetkillä syntyvät korkeataajuiset virtakomponentit oikosulkeutuvat heti suotokondensaattorien kautta. Näin virran kulkema silmukka on mahdollisimman pieni, joten se myös säteilee ympäristöönsä mahdollisimman vähän.

Komponenttivalintojen perusteella tehoasteesta piirrettiin piirikaavio, joka on esitetty liitteessä 1a.

4 Ohjausosan suunnittelu

Ohjausosio pitää sisällään kaiken sen elektroniikan, jota ei vielä käsitelty teho-osiossa, eli piensignaalien käsittelyn (moottorin parametrien tarkkailu, kortin lämpötila, käyttäjän käskyt), prosessoriympäristön, tehosignaalien ohjaukset sekä tulojännitteen reguloinnin em. osioiden haluamaan muotoon. Ohjausosan piirikaavio on esitetty liitteessä 1b.

4.1 Prosessoriyksikkö

Ohjausosion sydän on prosessori, joka muuntaa radiolta tulevan viestin PWM-signaaliksi kytkimille, hoitaa moottorin vaiheiden kommutoinnin ja varmistaa, ettei ylikuumenemista ja ylivirtatilanteita pääse tapahtumaan. Nykyisissä teollisuustaaajuusmuuttajissa käytetään monipuolisia ja kehittyneitä digitaalisia signaaliprosessoreita, jotka toimivat useiden kymmenten tai jopa satojen megahertsien taajuudella ja laskevat laskutoimituksia sekä kokonais- että liukuluvuilla hyvin nopeasti. Tässä työssä päätettiin kuitenkin selvittää huomattavasti kevyemmällä strategialla, harjaton DC-moottori kun on huomattavasti helpompi ohjattava kuin varsinaiset vaihtosähkömoottorit, eikä säädön tarkkuuskaan harrastuskäytössä ole kriittinen. Koska kokemus on osoittanut, että prosessorin ominaisuuksien loppuminen kesken projektin kuluttaa erittäin paljon aikaa, rahaa ja hermoja, valittiin työhön kuitenkin Microchipin varsin monipuolinen, erityisesti moottorinohjaukseen kehitetty PIC18F4431, jonka pääominaisuudet on esitetty taulukossa 4.1. Valittavaa kontrolleria puolsivat valmiiksi olemassaoleva kehitysympäristö sekä aikaisempi kokemus PIC16F-sarjan kontrollereista ja niiden ominaisuuksista. Tätä nykyä lähes jokaisella mikropiirivalmistajalla on oma kontrollerisarjansa, joten vaihtoehtoja kyllä löytyy. Tärkeimmät kontrollerivalmistajat etenkin kevyisiin moottorinohjaussovelluksiin ovat Microchipin lisäksi Atmel, SGS Thomson sekä Infineon.

Taulukko 4.1. Prosessoriksi valitun PIC18F4431:n ja sen sisarmallien pääominaisuudet.

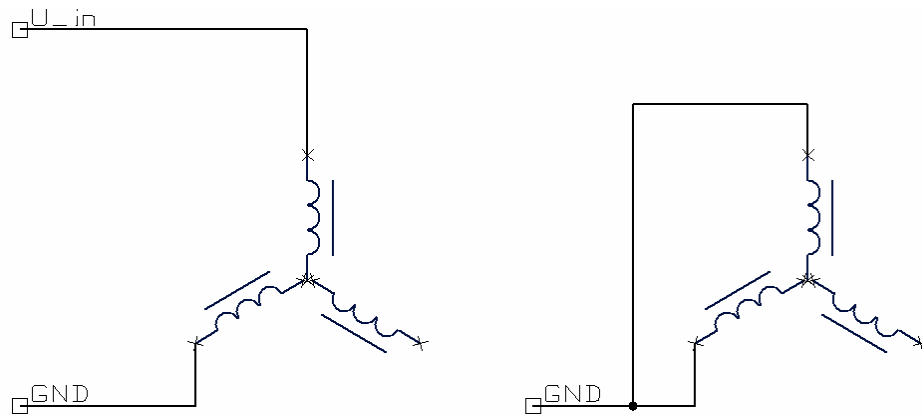
[<http://www.microchip.com>]

Device	Program Memory		Data Memory		I/O	10-bit A/D (ch)	CCP	SSP		EUSART	Quadrature Encoder	14-bit PWM (ch)	Timers 8/16-bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)				SPI	Slave I ² C™				
PIC18F2331	8192	4096	768	256	24	5	2	Y	Y	Y	Y	6	1/3
PIC18F2431	16384	8192	768	256	24	5	2	Y	Y	Y	Y	6	1/3
PIC18F4331	8192	4096	768	256	36	9	2	Y	Y	Y	Y	8	1/3
PIC18F4431	16384	8192	768	256	36	9	2	Y	Y	Y	Y	8	1/3

Valittu prosessori käyttää maksimissaan 40 MHz:n kellotaajuutta. Piirissä on sisäinen PLL eli vaihelukittu silmukka. Sen ansiosta on mahdollista käyttää kidettä, joka on vain neljäsosa piirin sisäisestä taajuudesta eli maksimissaan 10 MHz. Tämä vähentää kiteestä aiheutuvia suurtaajuisia häiriöitä, tai ainakin pitää ne matalammalla taajuusalueella. Pienemmänkin taajuuden käyttö on mahdollista, mutta tässä laitteessa haluttiin varmistaa maksimaalinen suoritusteho prosessorilta, minkä vuoksi kiteen tulee olla 10 MHz. Prototyyppiin kiteen koteloksi valittiin vanhanmallinen HC-49, joka on varsin kookas, mutta varsinaiseen lopputuotteeseen kide onkin helppo vaihtaa pienimpään mahdolliseen. Langattomien laitteiden, etenkin Bluetoothin, myötä hyvin pieniä kiteitä (jopa 3,2 x 2,5 mm) taajuusalueelle 4-20 MHz löytyy jo monelta valmistajalta.

4.2 Kytkimien ohjaus

Pulssisuhteen säätö laitteistotasolla toteutettiin siten, että pulssinleveysmodulaatiota käytetään vain virtaa syöttävässä vaiheessa, ja paluuvirran vaihe pidetään jatkuvasti maissa. Syöttävää vaihetta kytketään vuoroin maahan (PWM pois päältä) ja vuoroin käyttöjännitteeseen (PWM päällä). Näin kytkentähäviöitä tulee vain yhteen vaiheeseen kerrallaan, ja kytkevän vaiheen johtohäviöt minimoituvat, sillä kytkinten suojadiodit eivät pääse johtamaan.

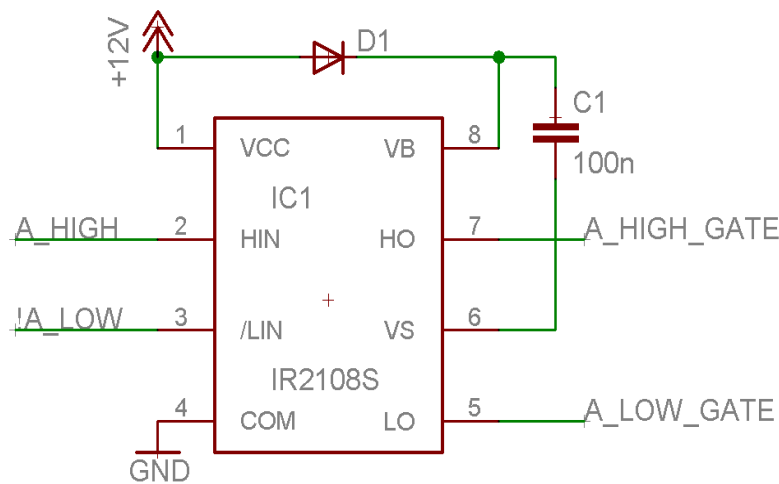


Kuva 13. Yhden jakson pulssisuhteen modulointi. Vasemmalla PWM päällä, oikealla poissa päältä.

Kytkimien ohjauspiiristö suorittaa ohjaussignaalin muokkauksen prosessorin antamasta pienivirtaisesta jännitetasosta sellaiseksi, jolla kytkimet aukeavat ohjauksen mukaan täysin ja riittävän nopeasti. Näin saadaan johtotilan sekä kytkentähetken häviöt minimoitua. Toisaalta liian jyrkkä kytkimen tilan vaihtaminen aiheuttaa suuria korkeiden taajuuksien pitoisuuksia virta- ja jännitemuotoihin, mikä saattaa aiheuttaa säteileviä ja johtuvia häiriöitä. Häiriöt heikentävät laitteen tai sen lähellä olevien laitteiden, etenkin radiovastaanottimen, toimintaa. Kätevintä on mitoittaa varsinainen ohjauspiiri mahdollisimman suurivirtaiseksi ja sitten rajoittaa kytkimen nopeutta hilavastuksella, joka hilakapasitanssin kanssa muodostaa alipäästösuodattimen. Hilavastus suojaa jossain määrin myös johtimien induktanssin aiheuttamalta soimiselta, joka saattaa aiheuttaa tuhoisan ylijännitepiikin kytkimen hilalle.

International Rectifier-yhtiön hilaohjauspiirit ovat laajalti käytettyjä, koska ne käyttävät vain vähän virtaa, ne ovat nopeita ja niillä on mahdollista ohjata suuriakin jännitetasoja. Piirit sisältävät kaiken puolisisillan ohjaukseen tarvittavan piiristön, joten käyttäjän tarvitsee lisätä vain ulkoinen bootstrap-diodi ja kondensaattori, joka diodin kautta latautuu alemman kytkimen (tai sen suojadiodin) johtaessa ja antaa ohjausjännitteen ylemmälle kytkimelle sitä kytkettäessä. Piirejä on saatavilla sekä yksi- että kolmivaiheisina, läpiladottavissa ja pintaliitoskoteloidissa, erilaisilla tulovaihtoehtoilla ja mahdollisella vikatilanteen lähdöllä. Tähän työhön valittiin IR2108S, SO-8-kotelossa oleva ohjain, jonka alempi sisäänvalo on vaiheenkääntävä ja ylempi suora ohjaus [17]. Tämä siksi ettei prosessori mahdollisessa vikatilanteessa ohjaisi sekä alempia että ylempiä kytkimiä yhtä aikaa päälle, olkoonkin että ohjauspiirin logiikan pitäisi kytkeä molemmat lähdöt pois päältä jos niitä yritetään yhtä aikaa ajaa päälle. Piirin

maksimilähtövirta on noin 200 milliampeeria, mikä ei ole kovin paljon, mutta riittää kyllä pienillä, parinkymmenen kilohertsin kytkentätaajuuksilla, joita tässä sovelluksessa käytetään. Piirin jännitekestoisuus riittää 600 voltin ohjaukseen, ja siinä on sisäänrakennettuna 540 nanosekunnin kuollut aika, jolla varmistetaan etteivät alempi ja ylempi kytkin johda yhtä aikaa tilan vaihtuessa. Työn teon aikana löydettiin myös National Semiconductorilta LM5100-sarja [18], joka on tarkoitettu pienemmän jännitteen ohjaimille (käyttöjännite alle 118 V), mutta pystyy antamaan huomattavasti suuremman maksimivirran (3 A). Lisäksi piiri pitää sisällään bootstrap-diodin, joten oheiskomponentteina tarvitaan vain yksi kondensaattori.



Kuva 14. Yhden vaiheen kytkinohjaimen kytkentä.

Jatkossa IR2108S on tarkoitus korvata diskreteilla puolijohteilla, sillä piirillä on minimikäyttöjännite 10,6 V. Kytkimet kuitenkin toimivat jo viidellä voltilla, ja toisaalta ohjainta saattaa olla tarve käyttää myös osana matalajännitteistä systeemiä, jossa tulojännite on alle seitsemän voltilla. Niinpä diskreteilla komponenteilla voidaan tehdä hilaohjauskytkentä, jonka alin toimintajännite ei ole näin korkealla. Vaadittavaa piirilevytilaa tuskin saadaan silti pienennettyä.

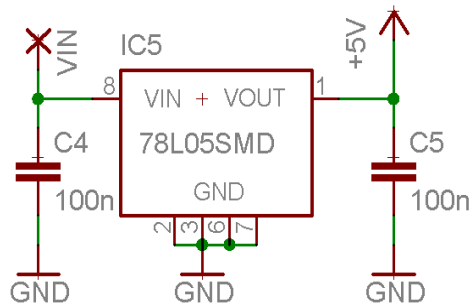
Kytkinohjainten ohjaus otettiin suoraan prosessorin kuudelta ohjelmoitavalta PWM-lähdeltä. Käyttökokemusten pohjalta harkittiin kuitenkin ohjauksen toteuttamista jatkossa siten, että prosessori antaisi vain kuuden kytkimen tilat sekä yhden pulssinleveystiedon, joka vaikuttaa kulloinkin syöttävään vaiheeseen suoraan laitteiston kautta. Näin säästytäisiin vaatimukselta moneen PWM:ään, mikä puoltaa kallista

prosessoria. Lisäksi PIC18F4431:n PWM-systeemi on hieman liian hienostunut siten, että kommutointivaiheessa prosessori vaatii monta käskyä vaiheen vaihtamiseksi ja pulssisuhteen päivittämiseksi. Tämä näkyy myös suoritusajassa, tosin ulkoinen logiikka prosessorin ulkopuolella aiheuttaa toki myös oman etenemisviiveensä ja kasvattaa piirilevyn pinta-alaa.

4.3 Jänniteregulointi

Prosesessori tarvitsee käyttöjännitteekseen maksimissaan viisi voltia, jonka voi joko ottaa vastaanottimen akusta tai reguloida moottorin syöttöjännitteestä. Mikäli käytetään jälkimmäistä tekniikkaa, on mahdollista tehdä regulaattorista niin tukeva, että se kestää ns. BEC-käytön (Battery Eliminator Circuit). BEC-käytössä vastaanottimen akku voidaan jättää lennokin kyydistä kokonaan pois ja radiolaitteita syöttää ajoakusta. Tämä säästää painoa, mutta ajoakun kapasiteetin loppuessa kone jää täysin ohjauskyvyttömäksi. Lisäksi regulaattori vie usean komponentin vuoksi paljon piirilevytilaa (hakkuritekniikalla) tai synnyttää paljon hukkalämpöä (lineaaritekniikalla). Toinen, suuremmissa teholuokissa käytetty tekniikka on erottaa radiolaitteisto optisesti ajopiiristä, jolloin häiriöiden kytkeytyminen herkän radion ja suuritehoisen moottorin välillä minimoituu. Työssä päätettiin käyttää ensisijaisesti jälkimmäistä tekniikkaa, jolloin prosessoria varten täytyy reguloida vain melko pieni virta. Lisäksi kytkimien ohjausta varten tarvitaan jännite väliltä 10-20 V.

Kummatkin jännitereguloinnit toteutettiin geneerisillä 7805- ja 7812-lineaariregulaattoreilla [19] SO-8-pakkauksessa, joka ei ole pakkauksena tässä tarkoituksessa pienin mahdollinen, mutta jonka vaatima piirilevytila on koko lailla sama kuin pienemmilläkin koteloilla tai diskreteilla komponenteilla toteutetulla piirillä. 78xx-sarjan piireille luvataan käytetyssä kotelossa 100 mA lähtövirta, minkä pitäisi riittää prosessorin, signaalinkäsittelyn ja kytkinohjauksen tarpeisiin.

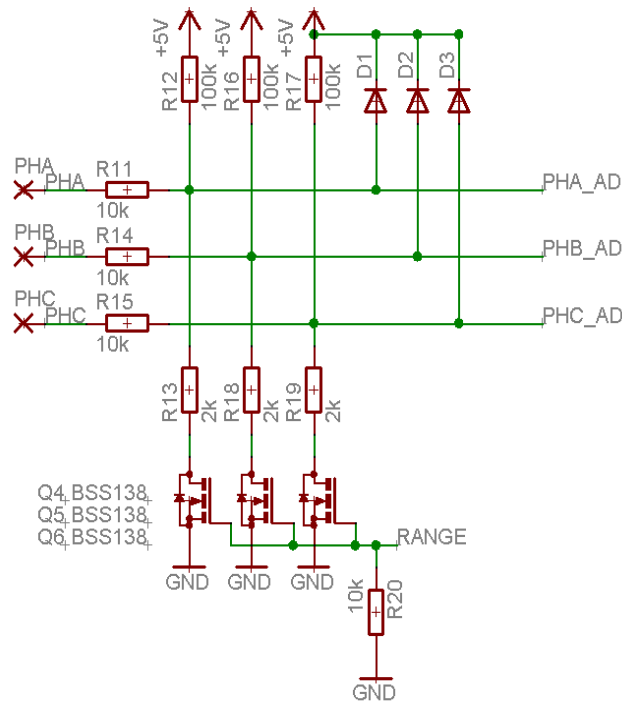


Kuva 15. Lineaarinen jänniteregulointi tulosähköstä viiteen volttiin.

Myöhemmissä protoissa 7805-piirin voisi mahdollisesti korvata Texas Instrumentsin piirin TPS5430 [20] ympärille rakennetulla hakkurilla. Kyseinen piiri antaa jopa kolme ampeeria lähtöönsä SO-8-kotelossa, joten se riittäisi hyvin myös BEC-käyttöön.

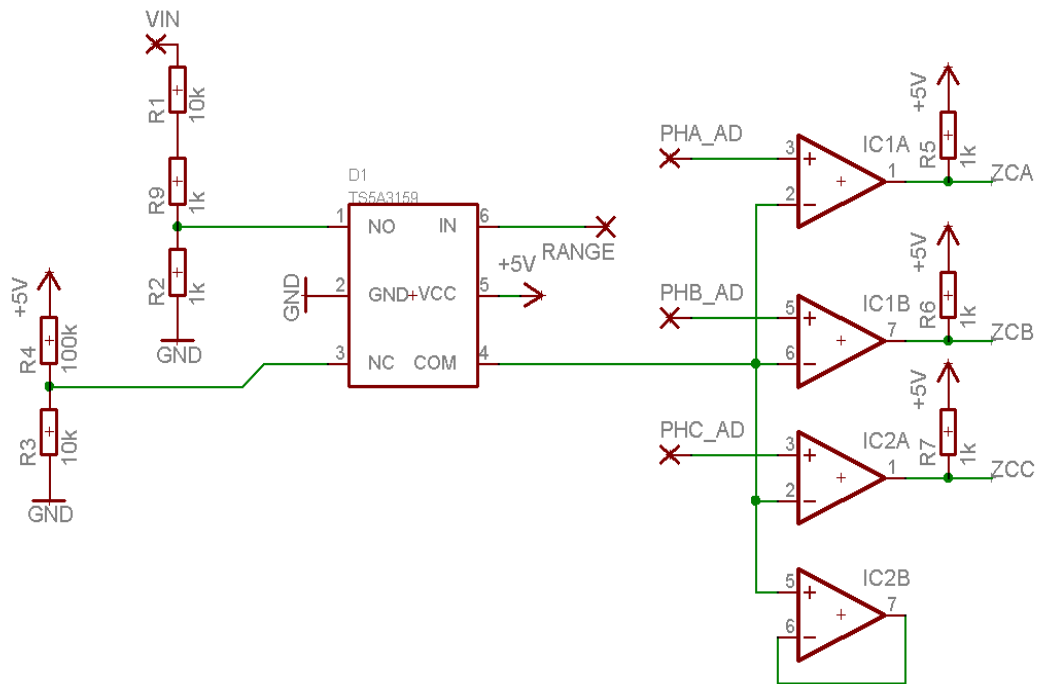
4.4 Vaihejännitteiden mittauss

Vaihejännitteiden normaali käyttöalue tässä työssä kattaa jännitealueen noin -0,5...30 V. Jotta jännitteet ovat helposti käsiteltävissä, tulee ne ensin skaalata mikrokontrollerin jännitealueelle eli 0...5 V. Skaalaus päätettiin toteuttaa luvussa 2.2.3 esitellyin metodein siten, että alhaisilla pulssisuhteen arvoilla mitattavat jännitteet leikkautuvat yläpäästään mutta vaimenevat alapäästään mahdollisimman vähän, ja suurilla pulssisuhteen arvoilla jännitteet taas skaalautuvat suoraan kontrollerin jännitealueelle. Kytkentä on helppo toteuttaa vain kolmella vastuksella ja yhdellä kytkimellä vaihetta kohden. Kaksi vastusta siirtää negatiiviset jännitteet positiiviselle alueelle ja kolmas vastus sekä kytkin vaimentavat tarvittaessa signaalin amplitudia. Aiemmin mainittu yläpään leikkautuminen järjestyy mittaavan piirin sisäisillä suojadiodeilla, jotka johtavat maasta signaaliin ja signaalista käyttöjännitteeseen. Vastusten mitoitus taas lähtee siitä, että suurimman mahdollisen jännitteen leikkautuminen käyttöjännitteeseen ei aiheuta kovin suurta virtaa signaalista käyttöjännitteeseen. Tässä työssä maksimiksi valittiin 2,5 mA. Muut vastukset määräytyvät suhteessa tähän etuvastukseen. Vaimennuskytkimet voivat olla lähes mitä vain, kunhan niiden päästötilan jännite on melko pieni, jottei signaali häiriinny. Tässä työssä päätettiin käyttää piensignaalfettejä.



Kuva 16. Vaihejännitteiden skaalauskytkentä. PHA, PHB, PHC ovat moottorilta tulevat vaihejännitteet, PHA_AD, PHB_AD, PHC_AD ovat käsiteltyt jännitteet valmiina vertailtavaksi referenssitasoon. Aktiivinen RANGE-signaali vaimentaa vaihejännitteet kuudesosaansa.

Vaihejännitteiden vertailu referenssitasoihin voidaan toteuttaa usealla tavalla. Helpoimmat tavat ovat joko analoginen vertailu komparaattorikytkennällä tai vaihtoehtoisesti signaalien muunto digitaalseksi ja vertailun suorittaminen kontrollerilla. Analogi-digitaali-muunnoksen (A/D) käyttö vaatii vähemmän ulkoisia komponentteja, mikäli kontrollerissa on nopea muunnin sisäänrakennettuna, mutta ulkoinen komparaattorikytkentä on monissa tapauksissa huomattavasti nopeampi vasteeltaan. Tässä työssä kokeiltiin molempia tekniikoita ja havaittiin, että nollakohdan ylityksen tunnistuksen luotettavuus on komparaattorikytkennällä parempi. Komparaattoreilla myös suurin mahdollinen pyörimisnopeus on selvästi suurempi kuin A/D-muuntimella toteutetulla vertailulla.



Kuva 17. Vaihejännitteiden vertailu komparaattoreilla. Vaihtokytkimellä D1 valitaan käytettävä referenssitaso pulssisuhdealueen mukaan, joko nollataso tai käyttöjännitteen puoliväli. ZCA, ZCB ja ZCC ovat lähdöt kontrollerille.

4.5 Piensignaalien käsittely

Työhön valitun prosessorin monet lähdöt kykenevät ajamaan useiden milliampeerin kuormia, joten ledejä voidaan ohjata suoraan näistä lähdöistä ilman erillistä puskuripiiriä. Ledit vaativat kuitenkin kukin oman sarjavastuksen, ja koska ledejä on useita, kannattaa sarjavastusten olla lopullisessa laitteessa pintaliitosvastusverkkoja jotta piirilevytilaa säästyy. Muutenkin monissa paikoin tarvittavat yksittäiset vastukset kannattaa toteuttaa samalla resistanssilla, jolloin ne voidaan korvata vastusverkoilla. Laitteeseen toteutettiin kahdeksan ledilähtöä, joilla voidaan seurata ohjelman suoritusta ja mahdollisesti myös parametrejä.

PIC-prosessoreilla on kaksi tuloa/lähtöä, joita käytetään myös ohjelmointiin. Ohjelmointilaitteen pitää pystyä ohjaamaan niitä, eli käytännössä nämä signaalit on parasta sovelluksessakin pyrkiä pitämään tuloina prosessorille, jotta impedanssitaso pysyy riittävän korkeana. Varmuuden vuoksi nämä tulot jätettiin vain ohjelmoinnin käyttöön.

Kortille toteutettiin myös puskuripiiri sarjaliikennettä varten, sillä lähes kaikissa PIC-proessoreissa on yksi tai useampi sarjaliikennelähtö. Sarjaliikenteellä laitteen tietoja voidaan lukea tietokoneelle ohjelman suorituksen aikana, jolloin toiminnan varmistaminen on joissain tapauksissa helpompaa kuin ledien tai oskilloskoopin avulla. Prosessorin sarjaliikennelähdön taso on nolasta viiteen volttiin ja RS-232-standardin mukainen signaalitaso on +15/-15 voltia, ja puskuripiiri hoitaa tämän muunnoksen.

5 Ohjelmistokehitys

Laitteen ohjelmisto kirjoitettiin C-kielellä käyttäen Custom Computer Services Inc. -yhtiön PCWH-kääntäjää, Microchipin MPLAB-ohjelmistoa ja ICD-2-ohjelmointilaitetta. Yhdistelmä oli nopea ja helppo käyttää, sillä uuden ohjelmistoversion kääntäminen ja kirjoitus prosessorille kestää vain noin kymmenen sekuntia. Ohjelmiston olisi voinut kirjoittaa myös suoraan prosessorin omilla käskyillä Assembler-kielellä, jolloin ohjelmiston suorittamisen nopeutta olisi ollut helpompi valvoa ja parantaa. Tällöin kuitenkin ohjelmistokehitys olisi ollut hitaampaa ja koodi olisi ollut vaikeaselkoisempaa kuin yleisesti ymmärretty C-kieli.

Ohjelmistokehitys toteutettiin pienissä erissä siten, että mahdolliset ongelmat tulisivat mahdollisimman aikaisessa vaiheessa esiin. Jokainen yksittäinen toiminto testattiin aluksi yksittäisenä, ja hiottiin toimivaksi ennen kuin varsinaista laitteen toimintaa alettiin hahmotella. Tässä käydään läpi laitteen toiminnan kannalta tärkeimmät lohkot, eikä niinkään yksittäisiä funktioita. Valmis ohjelmisto on esitetty liitteessä 2.

5.1 Laitteen alustus

Tärkeysjärjestyksessä ensimmäinen asetus laitteen käynnistyttyä on säätää PWM-lähdöt tehoasteelle inaktiiviseen tilaan, sillä satunnainen tila näissä lähdöissä voi johtaa helposti ja nopeasti tehoasteen tuhoutumiseen. Tämän jälkeen määritellään mitkä prosessorin lähdöistä ovat tuloja ja mitkä lähtöjä. Kolmas alustuskohde ovat laitteen sisäänrakennetut laskurit.

Päälaskuri eli Timer0 asetettiin suurimmalle mahdolliselle taajuudelleen, joka on piirin kellotaajuus jaettuna neljällä eli 10 MHz. Jokaisen askeleen kesto on siis 0,1 μ s ja koska Timer0 on 16-bittinen, se vuotaa yli 6,5536 millisekunnin välein. Askelkoko antaa enemmän kuin riittävän resoluution muun muassa servopulssin mittaamiselle.

Timer1 valittiin laskuriksi, jonka pohjalta kommutointi suoritetaan. Laskuri on 16-bittinen kuten Timer0:kin, mutta Timer1:n taajuus päätettiin asettaa neljäsosaan

maksimista. Näin täysi laskurikierrös kestää 262,144 ms, jolloin täydellä laskulla per vaihe saadaan sähköiseksi pyörintänopeudeksi

$$n = \frac{60 \frac{\text{s}}{\text{min}}}{0,262144\text{s} \cdot 6} \approx 38 \frac{1}{\text{min}} \quad (6)$$

joka on erittäin alhainen normaalia käyttötilannetta ajatellen.

Timer2 hoitaa PWM:n ohjearvon päivityksen ajastuksen. Sopiva jakso tälle on esimerkiksi noin 400 us, mikä saadaan aikaan asettamalla askeleen pituudeksi 1,6 us ja käyttämällä Timer2:n laskun pituuden asetusta hyväksi. Laskun pituudeksi asetettiin 249. Kun laskuri saavuttaa tämän arvon, se nollautuu automaattisesti.

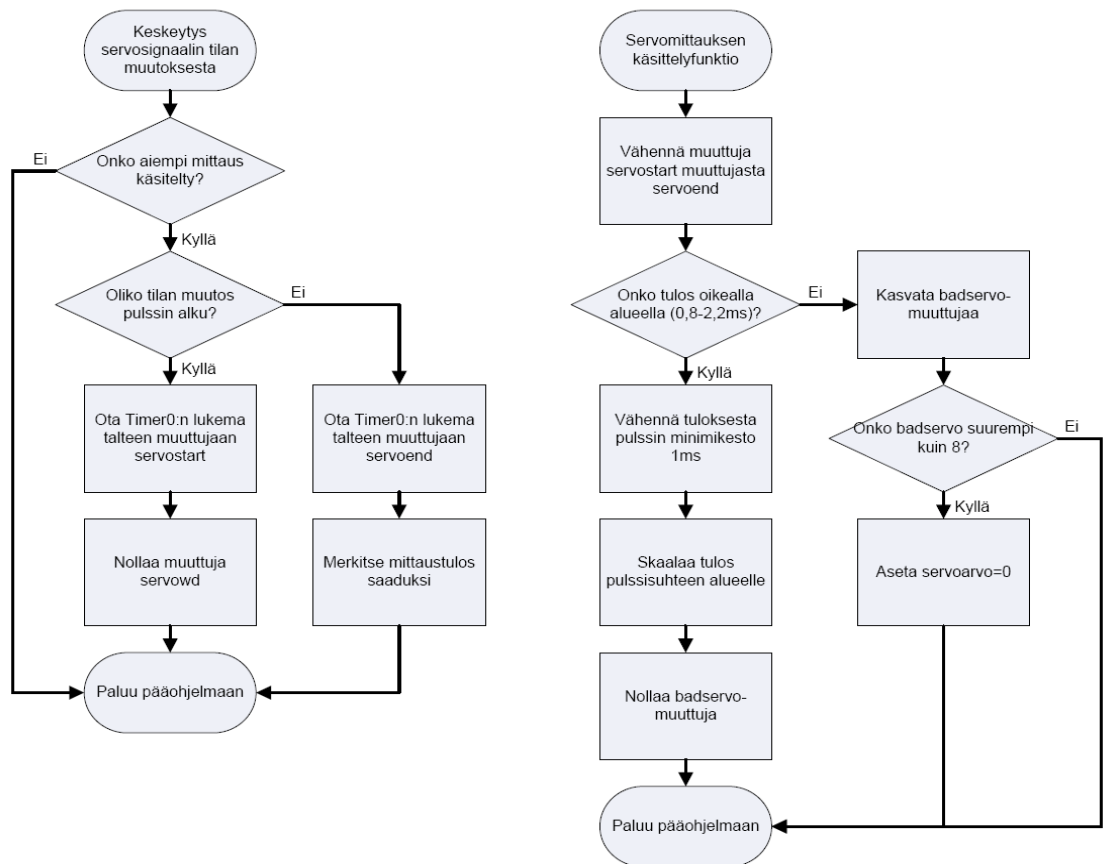
PWM-osio pitää kytkeä toimimaan halutulla tavalla. Tähän sisältyy lähtöjen määrän valinta, lähtöjen polariteetti ja PWM:n taajuus sekä muodostustapa. PIC18F4431 tarjoaa useita eri vaihtoehtoja PWM:n muodostukselle perinteisen ylöspäin laskevan laskurivertailun lisäksi. Vaihtoehtoisilla tavoilla voidaan vähentää tehoasteen aiheuttamia häiriöitä levittämällä kytkennöistä syntyvän häiriösignaalin spektriä tasaisemmin taajuuskaistalle. Tässä työssä kuitenkin perinteinen tapa riitti. Taajuudeksi valittiin juuri korvan kuulemattomissa oleva 20 kHz. Lähtöjen polariteetti määräytyi ohjainpiirien mukaan (alemmat kytkimet aktiivisena alhaalla, ylemmät ylhäällä) ja lähtöjen määrä oli jokaiselle vaiheelle ylä- ja alapää eli kuusi lähtöä. Pulssisuhteen säätöalueeksi tulee 20 kHz:llä 0...2000.

AD-muunnin säädettiin konvertoimaan virtamittaustuloa jatkuvasti ja antamaan keskeytys aina muunnostuloksen valmistuessa. Muunnoksen kellotaajuus säädettiin melko alas, sillä tuloksia käsitellään vain Timer2:n määrittelemän välein (noin 400 us).

Viimeinen alustustoiminto on haluttujen keskeytysten salliminen. Keskeytykset keskeyttävät perusohjelman ajamisen ja siirtyvät suoraan ennalta määrättyyn keskeytysrutiiniin, joten niitä käytetään lähinnä aikakriittisten toimintojen suorittamiseen. Suunniteltu moottoriohjain tarvitsi yhteensä viisi keskeytystuloa, jotka ovat yllä mainitut kolme laskuria Timer0, Timer1 ja Timer2, ulkoinen keskeytys servopulssilta ja AD-muunnoksen tulos. Lisäksi PICissä pitää asettaa bitti, joka määrittelee käytetäänkö keskeytyksiä lainkaan.

5.2 Servopulssin mittaaminen ja käsittely

Helpoin tapa mitata servopulssin kesto on käynnistää laskuri nolasta kun pulssi alkaa, ja ottaa lukema ylös kun pulssi loppuu. Tämä tarkoittaisi kuitenkin, että laskuri soveltuisi vain ja ainoastaan servopulssin mittaamiseen, sillä muut funktiot eivät voi luottaa siihen, että laskuria ei välillä ole nollattu servopulssin alkavan reunan vuoksi. Parempi tapa on antaa yhden laskurin laskea vapaasti koko lukualueensa, ja ottaa sekä pulssin alku- että loppuhetkeltä laskurin lukema ylös. Pulssin kesto on yksinkertaisesti näiden erotus. Servopulssin tilan vaihtelun tarkkailu kannattaa hoitaa keskeytyksillä, jotta laskurin lukema saadaan mahdollisimman nopeasti talteen ilman että suuri osa prosessorin tehosta hukkuu tilan vaihtumisen odottamiseen. Ohjelmistoon servopulssin mittausta ja sen skaalaus PWM-ohjeen lukualueelle toteutettiin osittain keskeytyksin, osittain pääohjelmassa. Keskeytyksellä hoidetaan laskurien lukeman muistiin ottaminen eli mittausta, ja skaalaus hoidetaan vasta pääohjelmassa niinä ajanhetkinä kun vaiheiden ohjauksen kanssa ei ole kiire, eli juuri kommutoinnin tai nollakohdan ylityksen jälkeen. Servopulssin mittaamisen ja käsittelyn vuokaavio on esitetty seuraavan sivun kuvassa.

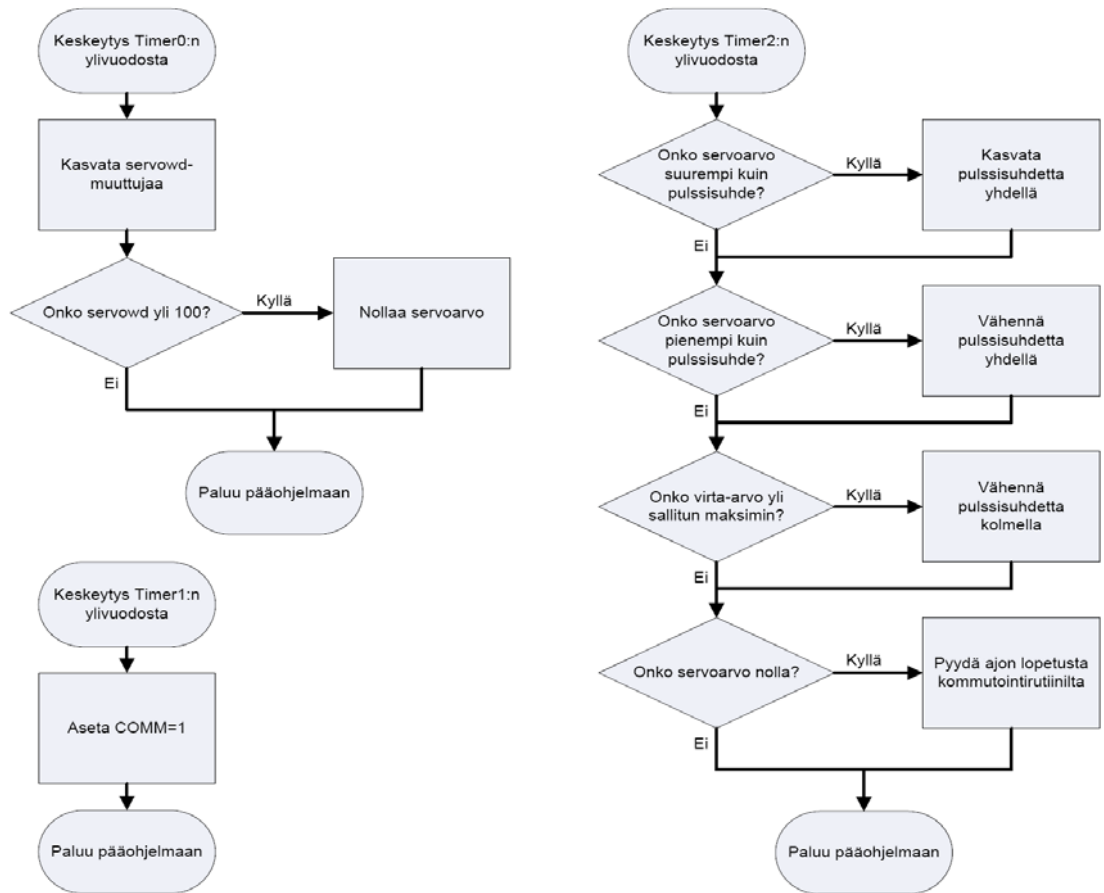


Kuva 18. Servopulssin mittaamisen ja käsittelyn vuokaavio.

Tässä työssä toteutettiin pulssisuhteen säätö siten, että pulssisuhde seuraa ohjettaan rampilla, jonka jyrkkyys on säädetty vakioksi. Mikäli virta ylittää ohjaimen maksimin, alennetaan pulssisuhdetta jonkin verran suuremmalla rampilla. Käytännössä aina, kun laskuri Timer2 pyörähtää ympäri (400 us välein), suoritetaan pulssisuhteen vertailu ohjeeseensa, ja sen jälkeen virran vertailu maksimiinsa. Ero pulssisuhteessa ja sen ohjeessa kasvattaa/vähentää pulssisuhdetta yhdellä jokaisella Timer2:n syklillä, ja virran ylittäessä maksimiinsa pulssisuhdetta vähennetään kolmella jokaisella Timer2:n syklillä. Jotta moottorin voi pysäyttää nopeasti, toteutettiin samaan yhteyteen myös funktio, joka kertoo pääohjelmalle mikäli ohjaus on nollassa. Tällöin pääohjelmasta siirrytään suoraan pysäytysrutiiniin.

Timer0 on valjastettu toimimaan myös servopulssin tarkkailijana: Aina kun servopulssin mittaustulos käsitellään, nollataan vahtikoiramuuttuja. Timer0 kasvattaa tätä muuttujaa jokaisella syklillään yhdellä, ja mikäli vahtikoira ylittää asetetun arvon (työssä käytettiin sataa eli noin 0,66 s), säädetään pulssisuhteen ohjearvo nolnaan. Näin

servopulssin katoaminen esimerkiksi radiohäiriön vuoksi pysäyttää moottorin pienen odotusajan jälkeen.



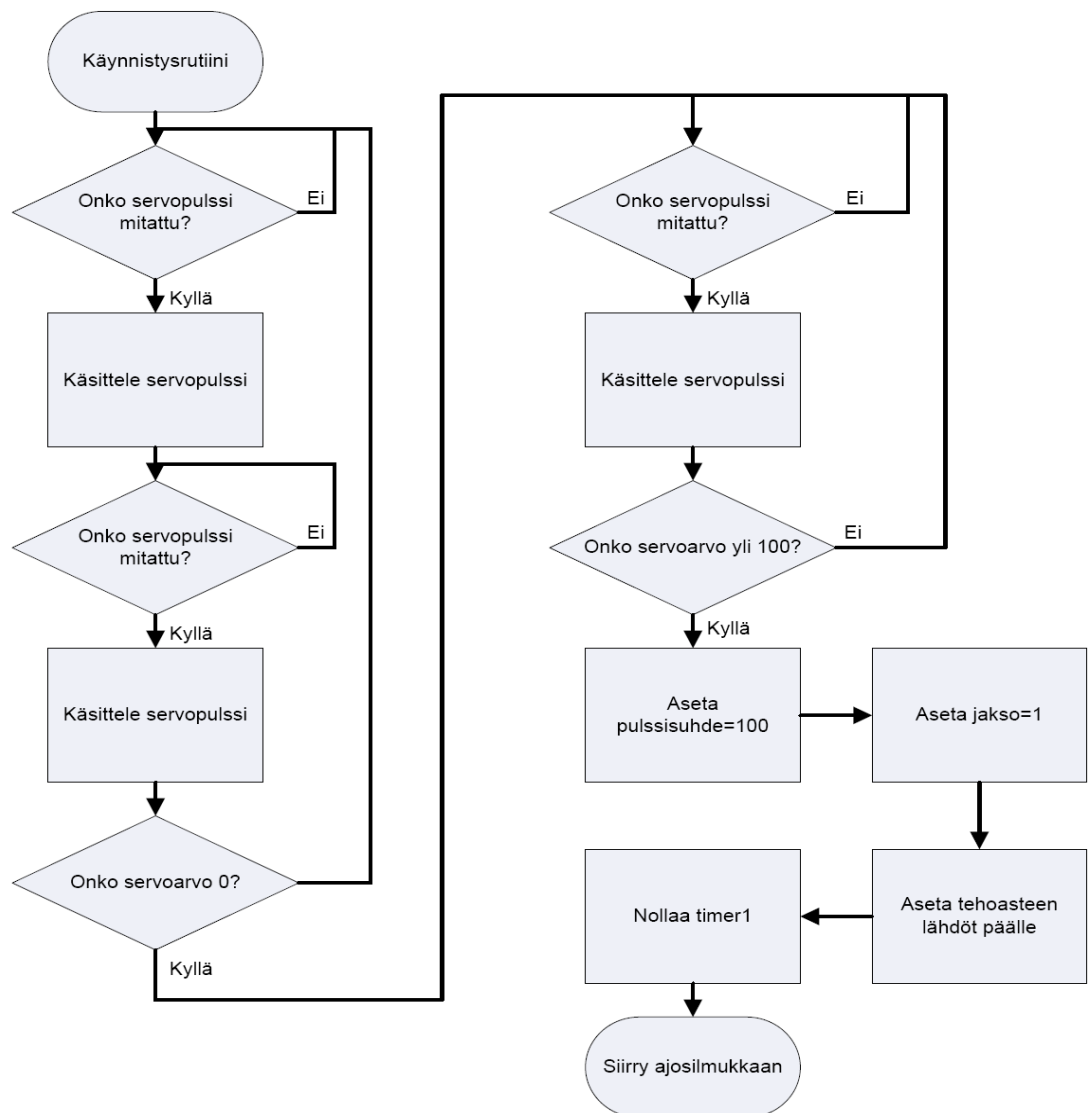
Kuva 19. Laskureiden keskeytysrutiinit.

5.3 Virran mittaus

AD-muunnin tekee virtatiedosta jatkuvasti muunnoksia, ja tallentaa viimeiset neljä muunnostulosta. Näistä neljästä tuloksesta lasketaan virralle liukuva keskiarvo, jota käytetään virran maksimin ylityksen tarkkailuun. Sekä muunnosten tallennus että keskiarvoistus tehdään AD-muuntimen antaessa keskeytyksen. Muunnoksia ei ole tarve keskeyttää edes silloin, kun moottori pysäytetään.

5.4 Ohjauksen käynnistys

Ennen moottorin käynnistämistä ohjain tarkastaa, että pulssisuhteen ohjearvo on nollassa. Näin moottori ei lähde käyntiin kun laite käynnistetään vaikka radio-ohjaus olisi jäänyt väärään asentoon. Tämän jälkeen laite odottaa, kunnes ohjearvo saavuttaa vähintään arvon 100 eli pulssisuhteohje on 5%. Tämä siksi, että testaamisen aikana havaittiin moottorin käynnistyvän luotettavammin, kun pulssisuhte ei lähde kasvamaan nolasta vaan sillä on heti käynnistyksessä minimiarvo 5%. Käynnistys tämän jälkeen on vain ensimmäisen ohjausjakson päällekytkentä ja Timer1:n nollaaminen, jonka jälkeen siirrytään normaaliin kommutointiin.



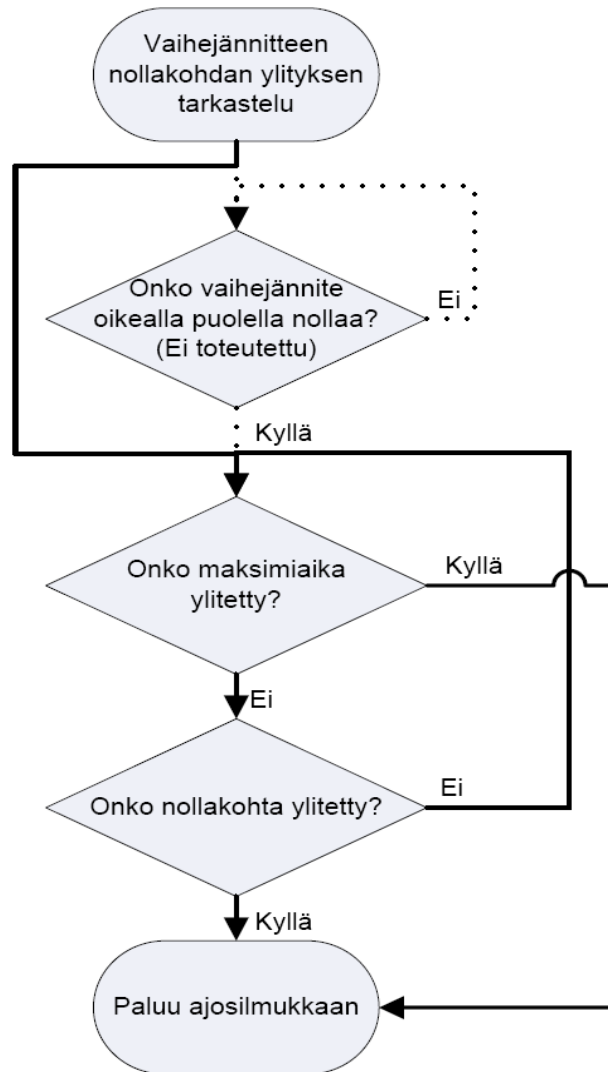
Kuva 20. Käynnistysrutiini.

5.5 Kommutoinnin ajoitus

Ajon aikainen pääohjelman silmukka oli alun perin yksi ja sama kaikille pulssisuhteen arvoille, mutta se päätettiin ohjelmiston kehityksen aikana jakaa eri silmukoiksi suurille ja pienille PWM:n arvoille, jolloin joitakin vertailuja voitiin jättää silmukoista pois, ja näin nopeuttaa silmukoita ohjelmamuistin kulutuksen kustannuksella. Kumpikin silmukka on kuitenkin toiminnaltaan yhtenevä.

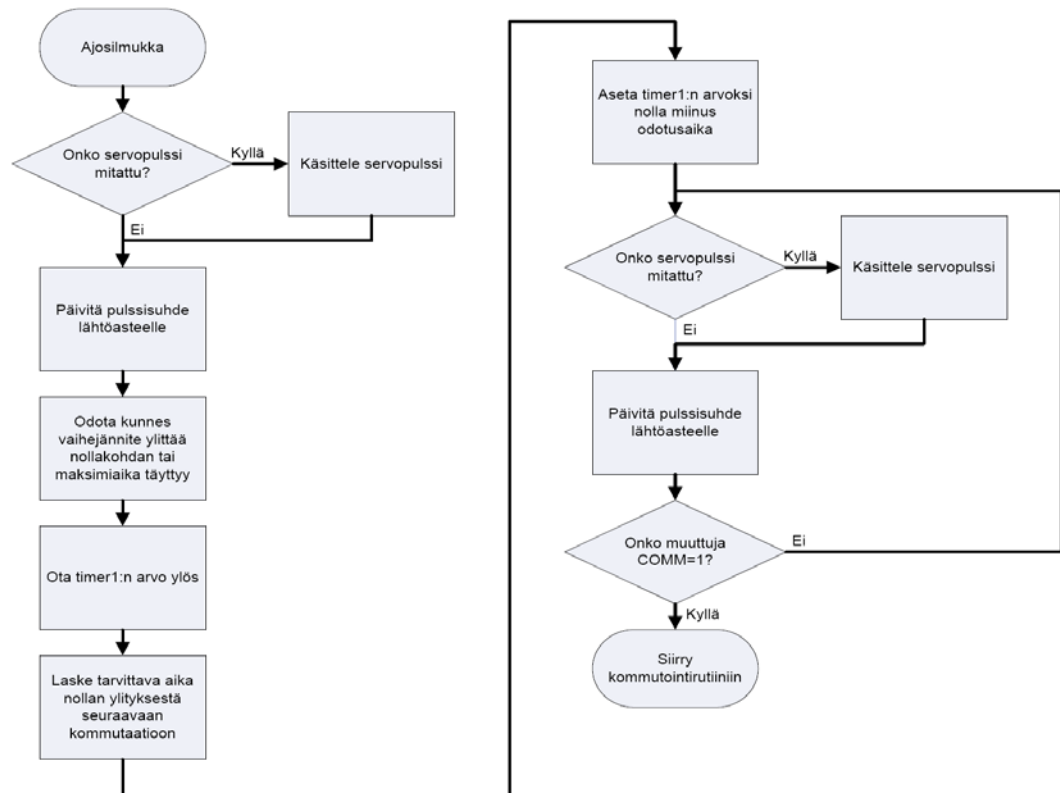
Ajosilmukka alkaa kohdasta, jossa kommutointi on juuri tapahtunut. Niinpä seuraavaan nollakohdan ylitykseen on pisin mahdollinen aika. On siis hyvä hetki laskea servopulssin kesto, mikäli servopulssi on saatu mitattua. Tämän jälkeen aloitetaan alisilmukka, joka odottaa nollakohdan ylitystä. Alisilmukassa tarkistetaan aluksi, onko kommutoinnista kulunut aika ylittänyt maksimin. Jos on, asetetaan Timer1:een arvoksi maksimiaika ja siirrytään odottamaan kommutointia. Muutoin tarkistetaan, onko nollakohta ylittynyt ja onko PWM-lähtö samalla hetkellä ollut oikeassa tilassa. Jos näin on, otetaan talteen Timer1:n arvo, eli aika kommutoinnista nollan ylitykseen, varmistetaan että nollakohdan ylityksen ilmaiseva signaali on edelleen oikeassa tilassa ja siirrytään laskemaan oikea aika nollakohdan ylityksestä kommutointiin. Muutoin palataan alisilmukan alkuun.

Ennen nollakohdan ylityksen tarkastelua olisi mahdollista odottaa, että vaihejännite käy nollan toisella puolella, jolloin pienellä nopeudella moottori pyörisi miniminopeuttaan kunnes vaihejännitteen amplitudi on tarpeeksi suuri. Käytännössä havaittiin kuitenkin, että tällainen silmukka aiheuttaa enemmän ongelmia kuin ratkaisee niitä: Moottorin ohjaus muuttui kokonaan epävarmaksi. Niinpä vaihejännitteen olemassaolon tarkastelusta luovuttiin: Ohjelma toteutettiin siten, että mikäli vaihejännite on heti tarkkailun alkaessa jo ylittänyt nollan, tilanne käsitellään normaalina nollan ylityksenä.



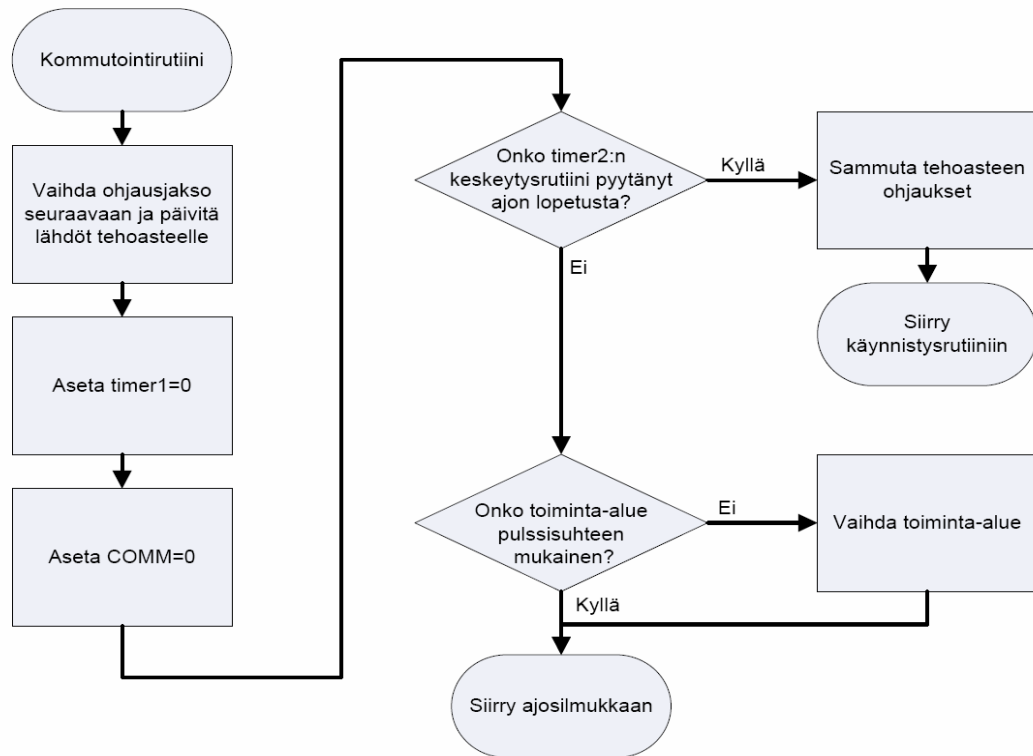
Kuva 21. Vaihejännitteen nollakohdan ylityksen tarkastelu.

Mikäli nollan ylitys havaittiin ajoissa (pyörimisnopeus yli minimin), lasketaan tarvittava viive ylityksestä kommutointiin, joka on yhtä kuin aika kommutoinnista nollan ylitykseen vähennettynä ohjelman suorittamisen aiheuttamilla viiveillä. Saatu arvo vähennetään nollasta ja asetetaan Timer1:een. Tämän jälkeen on taas hyvää aikaa käsitellä mahdollinen servopulssin mittaustulos. Ajosilmukan seuraava alisilmukka odottaa, kunnes Timer1 pyörähtää ympäri oikean ajan kuluttua. Timer1:n keskeytysrutiini asettaa ohjausbitin, jonka havaittuaan alisilmukka päästää ohjelman kulun kommutointirutiiniin.



Kuva 22. Ajosiilmukan vuokaavio.

Kommutointirutiini vaihtaa ohjausjakson seuraavaan ja säättää tehoasteen ohjauksen sen mukaiseksi. Tämän jälkeen Timer1 ja kommutointibitti nollataan, ja tarkistetaan onko PWM-ohjearvo nollassa. Jos ohjearvo on nolla, siirytään moottorin poiskytkentärutiiniin. Jos taas ollaan eri silmukassa kuin mitä PWM-arvo edellyttäisi, siirytään oikeaan silmukkaan. Muussa tapauksessa siirytään sen hetkisen ajosiilmukan alkuun.



Kuva 23. Kommutointirutiinin vuokaavio.

5.6 Moottorin poiskytkentä

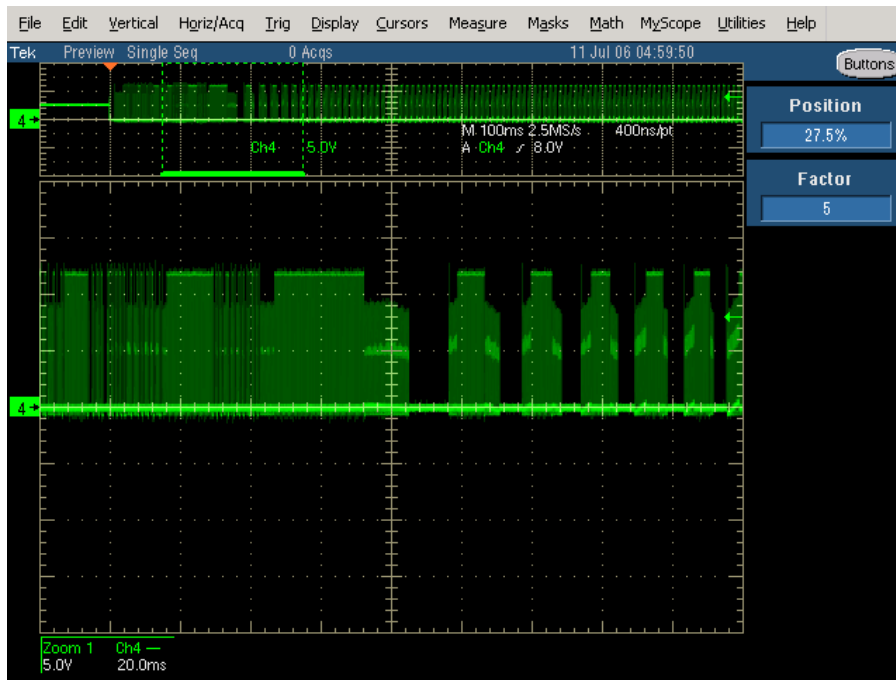
Moottorin poiskytkennän voi aiheuttaa kolme lähdettä: käyttäjän ohjaus, radiohäiriö jolloin servopulssi katoaa kokonaan tai radiohäiriö jolloin servopulssi on jatkuvasti poissa oikealta alueeltaan. Kaikissa tapauksissa moottori kytketään pois päältä mahdollisimman nopeasti. Jos aktiivinen jarrutus on valittuna ohjaimen parametreista, kytketään tehoasteen alemmat fetit kaikki päälle ja ylemmät pois päältä. Muuten kytketään kaikki fetit pois päältä. Tämän jälkeen siirrytään takaisin aloitusrutiinin kohtaan, jossa tarkistetaan että pulssinleveyden ohjearvo on nolla. Radiohäiriöiden aiheuttaman virhetilan voi siis kuitata käyttämällä ohjausta nollassa ja palauttamalla ohjaus sen jälkeen haluttuun tilaan.

6 Ohjaimen toiminta

Ohjaimen toiminta testattiin kahdella eri moottorilla: Toinen moottori oli vanhasta levyasemasta purettu moottori ilman ulkoista kuormaa, ja toinen varsinainen lennokokikäyttöön valmistettu puhallinmoottori. Lennokkimoottorin maksimivirta oli 8 A ja tahtinopeuden suhde jännitteeseen 3750 rpm/V. Testien aikana käyttöjännite oli pääasiassa 12 V, paitsi huippunopeustesteissä joissa jännitettä nostettiin kunnes ohjain ei enää pysynyt moottorin perässä.

6.1 Käynnistyminen

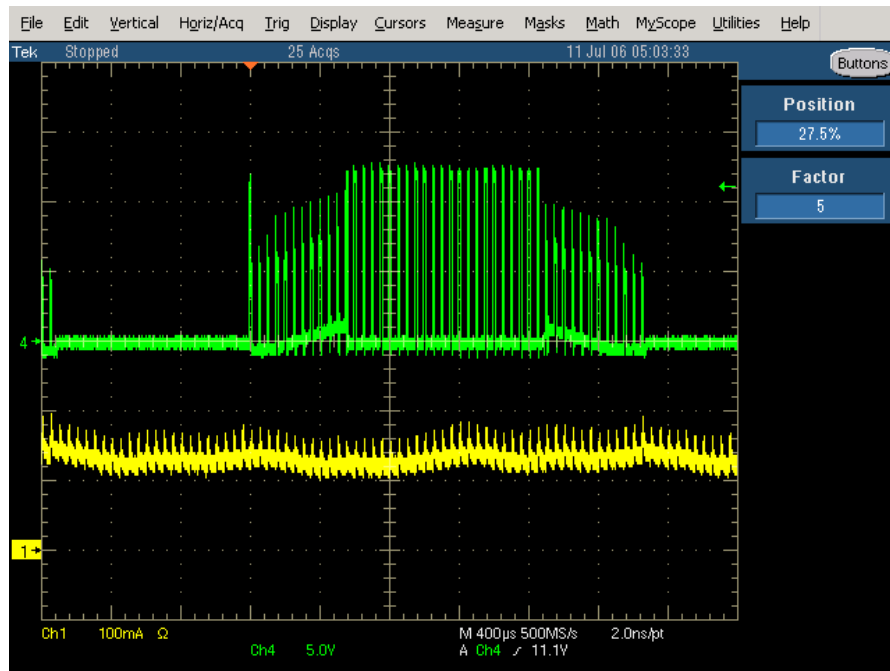
Testit aloitettiin nostamalla pulssisuhdeohjetta hitaasti. Aivan alhaisilla ohjearvoilla moottori alkaa täristä hallitsemattomasti, sillä tällöin vaihejännitteen amplitudi on lähes nolla, ja komparaattorit rekisteröivät pienimmätkin häiriöt nollakohdan ylityksinä. Tämä hallitsemattomuus ei kuitenkaan lennokokikäytössä haittaa, sillä potkureiden inertia on tyypillisesti pieni. Liikkeellelähtömomentti, ja siten myös käynnistysvirta, on tällöin myös pieni. Niinpä jo melko pieni pulssisuhde riittää antamaan moottorille sellaisen jännitteen, jolla virta ja siten vääntömomentti on tarpeeksi suuri kuorman pyörittämiseen. Tällöin vaihejännitteen amplitudi kasvaa riittävän suureksi ja ohjain siirtyy hallitulle alueelle. Hallitsematon alue ei siis ole haitallinen ohjaimelle, sillä käynnistyksessä ei tapahdu tilannetta, jossa yhtä aikaa lähtövirta olisi suuri ja kommutointitaajuus satunnainen. Jos näin kävisi, ohjaimen häviöteho saattaisi ylittää sallitut rajat ja jokin komponentti pettää. Alla olevassa oskilloskooppikuvassa nähdään käynnistyksen hetki, jolloin ohjain saa kommutoinnin takaisinkytkentälenkin suljettua ja moottorin ohjaus muuttuu hallituksi.



Kuva 24. Moottorin käynnistyksen aikainen yhden vaihejännitteen aaltomuoto. Suurennetussa osassa ensimmäiset 60-70 ms ovat suuritaajuista satunnaista kommutointia, minkä jälkeen moottorin ohjaus on hallittua (samanlaiset jaksot seuraavat toisiaan kiihtyvällä nopeudella).

6.2 Matalan pyörimisnopeuden alue

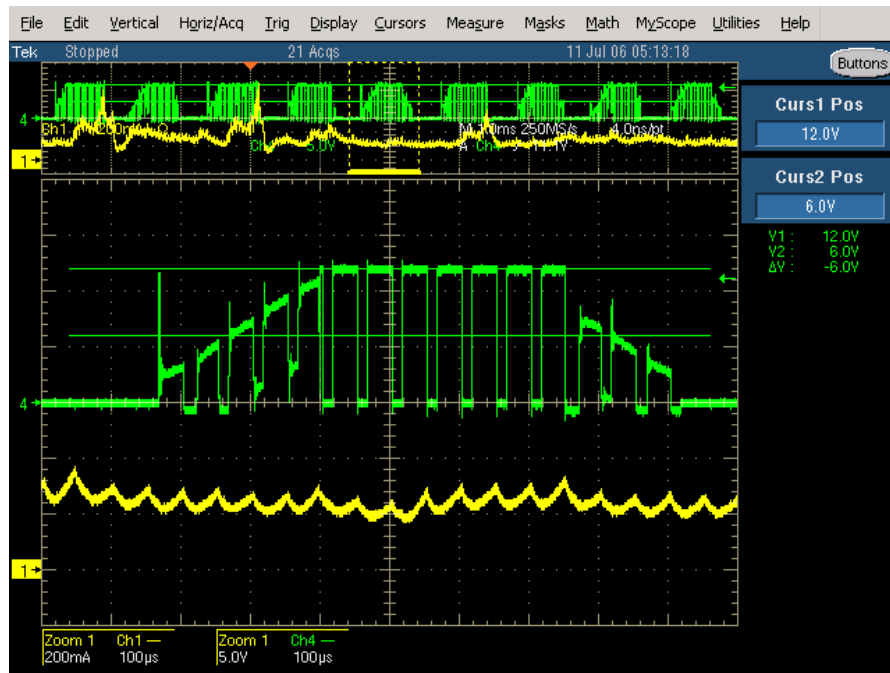
Toimiessaan hitaalla alueella ohjain tarkkailee hetkeä, jolloin inaktiivisen vaiheen indusoitunut jännite ylittää nollan PWM:n ollessa poissa päältä. Seuraavan sivun kuvasta nähdään, että ajoitus on melko kohdallaan: vaihejännite ylittää nollan inaktiivisten jaksojen keskikohdissa. Myös tasainen syöttövirta kertoo ohjauksen onnistuneen, sillä myöhästynyt tai liian aikainen kommutointi aiheuttaa ottovirtaan suuria vaihteluja.



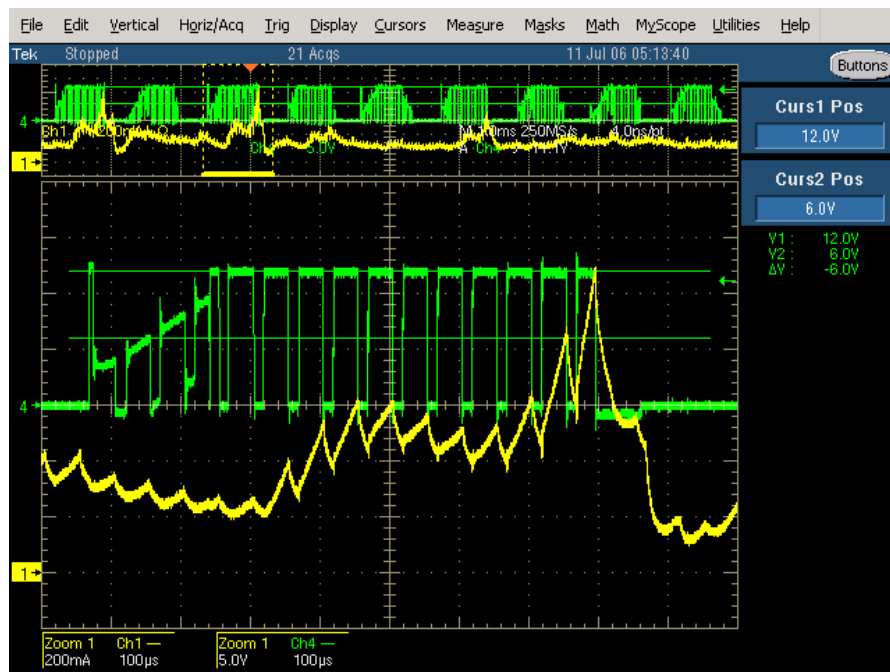
Kuva 25. Pienellä nopeudella vaihejännite yhden sähköisen kierroksen ajalta (CH4) ja syöttövirta (CH1). Vaihe on alussa noin 1,1 ms kytkettynä maahan, sitten noin 0,55 ms inaktiivisena, jonka jälkeen taas noin 1,1 ms kytkettynä säädetyllä pulssisuhteella käyttöjännitteen ja maan väliä ja viimeiseksi taas noin 0,55 ms inaktiivisena.

6.3 Korkean pyörimisnopeuden alue

Nopean pyörimisnopeuden alueella (pulssisuhde yli 50%) ohjain tarkkailee hetkeä, jolloin inaktiivisen vaiheen jännite ylittää käyttöjännitteen puolivälin PWM:n ollessa päällä. Suuremmalla nopeudella ohjaimen viiveet alkavat näkyä selvemmin. Vaikka alla olevassa kuvassa suurennettuna oleva sähköinen kierros onkin ajoitukseltaan melko hyvä, nähdään suurentamattomasta kuvasta virran vaihtelevan niin paljon, että ajoitus on varmasti jonkin verran epätarkka.



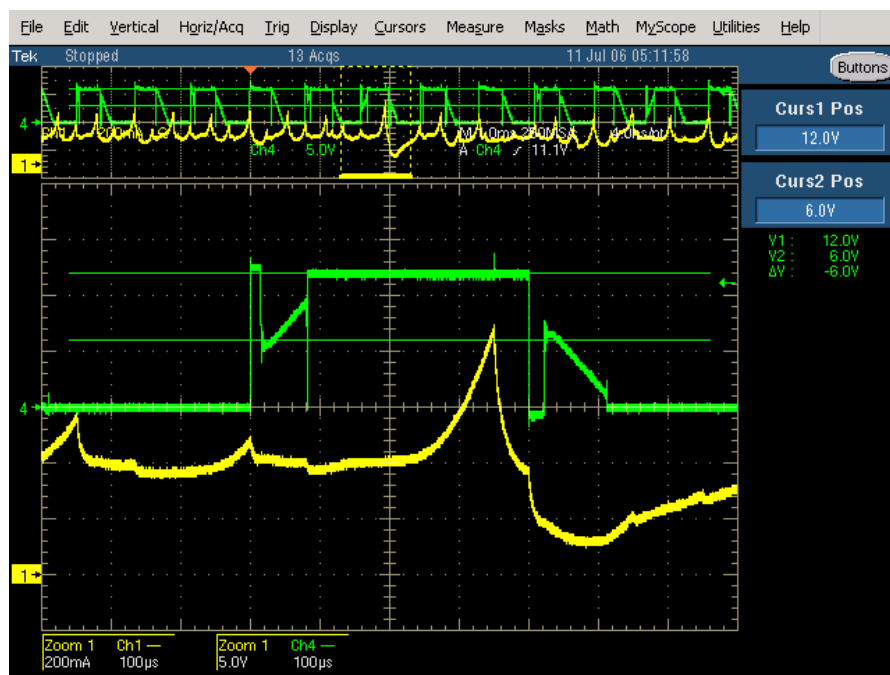
Kuva 26. Suurella nopeudella vaihejännite (CH4) ja syöttövirta (CH1) suurennettuna yhden sähköisen kierroksen ajalta, jolloin ajoitus on onnistunut. Suurentamattomasta mittauksesta (kuvan yläreuna) nähdään virran vaihtelevan rajusti muiden kierrosten aikana.



Kuva 27. Suurella nopeudella vaihejännite (CH4) ja syöttövirta (CH1) suurennettuna yhden sähköisen kierroksen ajalta, jolloin sekä virrasta että jännitteestä nähdään ajoituksen olevan pielessä.

6.4 Täyden pulssisuhteen toiminta

Kun pulssisuhde kasvatetaan maksimiinsa, poistuvat kytkemisten aiheuttamat pulssit vaihejännitteiden aaltomuodoista, ja muodoista nähdään helpommin kommutointien puoliväli. Merkittävänä seikkana nähdään tällöin myös moottorin hajainduktanssista aiheutuva jännitepulssi. Tämä pulssi syntyy, kun vaihe kytketään inaktiiviseksi mutta hajainduktanssi jatkaa virran kuljettamista vaiheessa, jolloin virta kulkee kytkinten suojadiodien lävitse joko vaiheesta positiiviseen kiskoon tai maakiskosta vaiheeseen. Jännitepulssin maksimikesto pitää ottaa huomioon kommutoinnin ajoituksessa, sillä muuten pulssi voidaan tulkita indusoituneen jännitteen nollakohdan ylitykseksi, mikä taas siirtäisi seuraavan kommutoinnin aivan liian aikaiseksi. Alla olevassa kuvassa on esitetty vaihejännite ja ottovirta kun pulssisuhde on sata prosenttia mutta ajoitus ei ole täysin onnistunut.

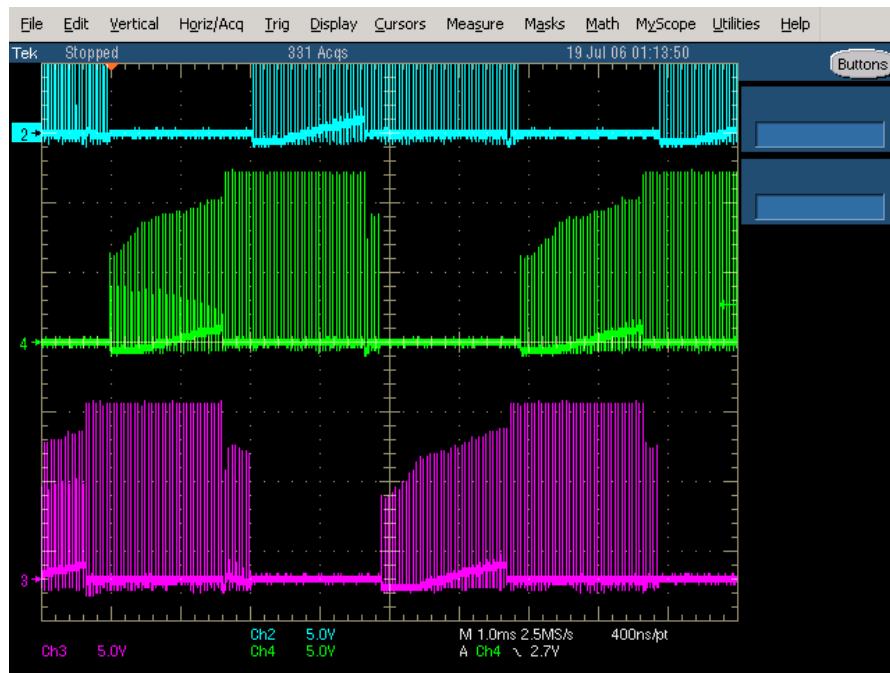


Kuva 28. Ohjaimen aaltomuodot (vaihejännite CH4, ottovirta CH1) täydellä pulssisuhteella suurennettuna yhden sähköisen kierroksen ajalta. Jännitemuodossa nähdään induktiiviset jännitepulssit inaktiivisten jaksojen alussa, virrassa nähdään jälleen ajoituksen pieni viivästys.

6.5 Kommutoinnin ajoituksen tarkkuus

Vaikka ohjaimen toiminnasta saakin eroteltua yksittäisiä sähköisiä kierroksia, joiden aikana ajoitus on oikeanlainen, näkyy ohjauksessa pidemmällä aikavälillä jatkuvasti ajoituksen epäonnistumista. Tarkkailtaessa kommutointeja suhteessa

vaihejännitekomparaattorien lähtöihin voidaan silti havaita, että ohjain toimii koko ajan oikein: komparaattori vaihtaa lähtönsä tilaa aina jakson keskellä. Virheajoitus joidenkin jaksojen kohdalla tuntuu korjautuvan melko hyvin taas seuraavan vaiheen kohdalla, joten keskimäärin moottori pyörii oikeanlaisesti. Virheelliset ajoitukset aiheuttavat kuitenkin ylimääräistä tehohäviötä, moottorin melutason kasvamista ja käyttöiän lyhenemistä. Todennäköisesti häiriöt kommutointitiheydessä johtuvat siitä, että ohjattavien testimoottoreiden vaihejännitteet ovat sinimuotoisia, mutta ohjain antaa trapetsoidaalista aaltomuotoa vaiheille. Useissa kaupallisissa säätimissä onkin mahdollista vaihtaa kommutoinnin ennakkoon siten, että kommutointi tapahtuu nollakohdan ylityksen jälkeen nopeammin kuin mitä kommutoinnista nollan ylitykseen kestää.

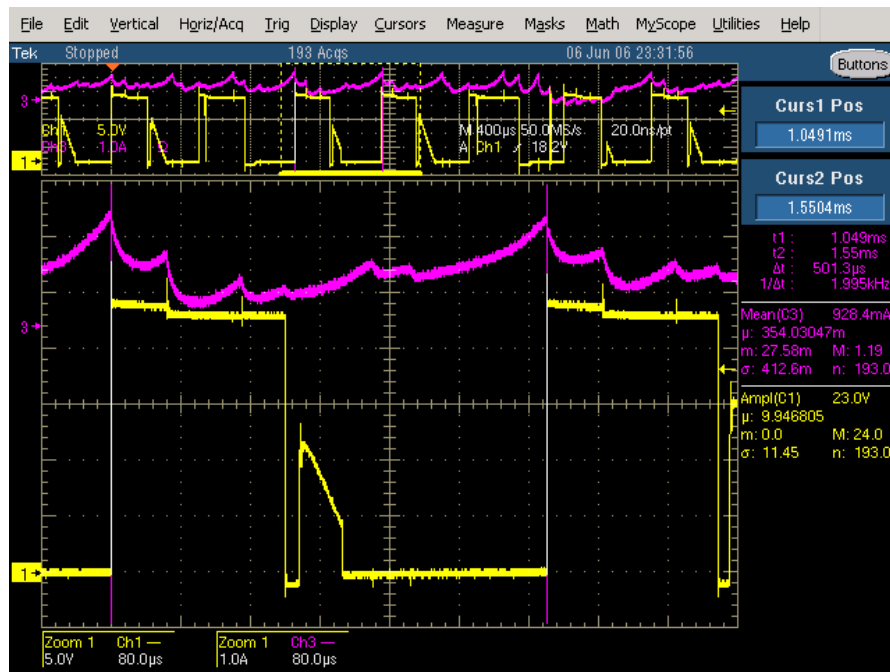


Kuva 29. Ohjaimen syöttämät vaihejännitteet, vaihe A (CH2), vaihe B (CH3) ja vaihe C (CH4). Jaksot, joissa indusoitunut vaihejännite on laskeva, ovat kaikki liian lyhyitä.

6.6 Ohjaimen maksiminopeus

Ohjaimen maksiminopeus testattiin kuormattomalla levyaseman moottorilla nostamalla jännitettä kunnes ohjain ei enää pystynyt seuraamaan moottorin pyörimisnopeutta. Jännite oli tällöin 23V ja sähköinen kierrosnopeus 120 tuhatta kierrosta minuutissa. Koska ohjain pystyi näinkin huimaan nopeuteen, on edelleen syytä epäillä että

kommutoinnin ajoituksen huojunta ei johdu ohjaimen hitaudesta vaan nimenomaan ohjattujen moottorien ominaisuuksista.



Kuva 30. Suurimman mahdollisimman nopeuden haku käyttöjännitettä nostamalla. Vaihejännite (CH1) ja ottovirta (CH3). Sähköinen nopeus on noin 2000 hertsiä eli 120 tuhatta kierrosta minuutissa.

6.7 Jatkokehitys

6.7.1 Laitteiston jatkokehitys

Jatkokehityksen tärkein yksittäinen kohde laitteiston kannalta on mikrokontrollerin vaihtaminen halvempaan ja ominaisuuksiltaan kevyempään versioon. Nykyinen kontrolleri on erittäin monipuolinen ja muunneltava, mutta monipuolisuuden mukana helpotkin yhden komennon tehtävät monimutkaistuvat usein monen käskyn ohjelmarutiineiksi. Sisäinen PLL on hyvä olemassa, jotta kellotaajuuden saa helposti 40 MHz:iin, mutta useimmat halvatkin kontrollerit pystyvät 20 MHz:in kellotaajuuksiin ilman PLL:ää. Kellotaajuuden puolittumisen taas ei pitäisi merkittävästi vaikuttaa ohjaimen toimintaan, sillä nykyiselläänkin ohjelma viettää suuren osan ajastaan erilaisissa silmukoissa odottaen tapahtumia, jotka havaittaessa jatkavat ohjelman suoritusta.

Laitteiston muita mahdollisia parannuskohteita ovat apujännitteiden generointi hakkurilla, minimikäyttöjännitteen alentaminen sekä komparaattorikytkennän nopeuttaminen. Viiden voltin apujännite voitaisiin toteuttaa tehokkaalla hakkurilla, jolloin tämän jännitteen voisi syöttää myös lennokin radiolaitteistolle. Näin radion oman akun voisi jättää pois. Minimikäyttöjännitteen alentaminen vaatii lähinnä tehokytinten hilaohjauspiirien toteuttamisen diskreeteillä komponenteilla. Haasteellista tässä on, että ohjauksen tulee olla hyvin luotettava, mutta se ei saa kuluttaa piirilevyalaa merkittävästi enempää kuin varsinainen hilaohjauspiiri. Komparaattorikytkennän nopeuden ja tarkkuuden optimointiin voisi vielä panostaa, sillä tarkempi vertailu mahdollistaisi käynnin vielä matalammalla nopeudella. Helpoin tapa vaikuttaa nopeuteen on vaihtaa komparaattorin tyyppiä, käytettyjen vastusten toleranssi taas vaikuttaa merkittävästi tarkkuuteen.

6.7.2 Ohjelmiston jatkokehitys

Kehitetyssä prototyypissä moottorin kommutoinnin ohjaus etenee suurelta osin lineaarisesti. Ohjelmistossa ainoastaan servopulssin mittausta, AD-konversion tuloksen talteenotto ja pulssisuhteen säätö tapahtuvat keskeytusrutiineina. PICin monissa kontrollereissa on mahdollisuus useisiin ulkoisiin keskeytyksiin, joten nollakohdan

ylitystieto vaihejännitteiden komparaattoreilta olisi hyvä kytkeä näihin tuloihin, jotta vaste havaintoon olisi mahdollisimman nopea. Näin ulkoisten keskeytystulojen ohjelmarutiini hoitaisi nollakohdan ylitykseen liittyvän käsittelyn ja ajastuksen, ja sopivan ajastimen keskeytysrutiini voisi hoitaa varsinaisen kommutoinnin.

Kommutoinnin jälkeen nollakohdan ylitystä odottaessa olisi myös mahdollista tarkistaa, että vaihejännite ensin käy oikealla puolella nollaa (ts. nousevalla rampilla vaihejännite käy ensin negatiivisella puolella ennen nollan havaitsemista ja laskevalla rampilla positiivisella puolella). Näin menetellen nollakohdan ylityksen havaitseminen olisi mahdollisimman nopeaa, ilman että induktiivinen jännitepulssi sotkee vertailua. Prototyypiversiossa kommutoinnin jälkeen on kiinteä 50 us tauko, jonka aikana vaihejännitettä ei tarkkailla, ja jonka aikana induktiivisen jännitepulssin pitäisi loppua.

Tärkeä jatkokehityskohde on lisätä laitteeseen mahdollisuus ohjata radiolaitteilla suoraan moottorin nopeutta eikä niinkään moottorin pulssisuhdetta. Tätä varten pitäisi kehittää ohjelmarutiini, joka esimerkiksi normaalissa pulssisuhdeohjauksessa kirjaisi ylös suurimman käytetyn kierrosnopeuden kullakin ajokerralla, ja kun käyttäjän ohjaus on nollassa, siirtyisi ohjain nappia painamalla nopeusohjatuksi siten, että käyttäjän antama maksimiohjaus vastaisi ylöskirjattua huippunopeutta. Näin nopeusskaala olisi helposti ja nopeasti skaalattavissa halutuksi. Nopeussäätö olisi myös helppo toteuttaa rinnakkain pulssisuhteen säädön kanssa, jotta käyttäjä voi valita ohjaimeen sopivan toimintatilan käyttökohteesta riippuen.

Jatkossa AD-muunnokset olisi hyvä synkronoida PWM-taajuuteen, sillä siten pulssimodulaation aiheuttama rippeli syöttövirrassa eliminoituisi mittauksista, ja virran säätäminen olisi helpompaa. Virran tarkkailulla voitaisiin myös havaita mahdolliset liian pitkäksi venyvät kommutointien välit, mikä osaltaan tasoittaisi ottovirtaa ja moottorin pyörimisnopeutta.

7 Johtopäätökset

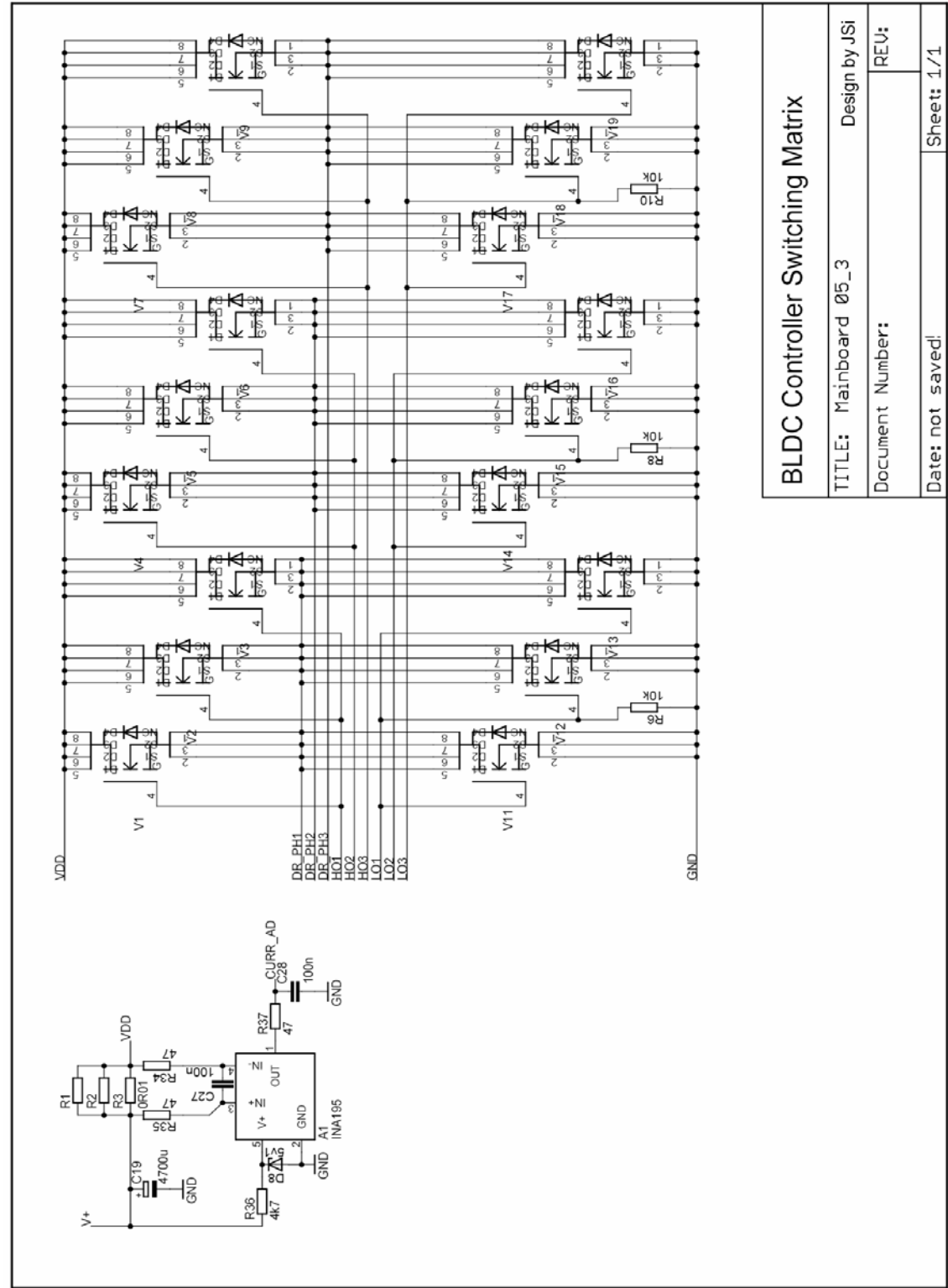
Suunniteltu harjattomien tasavirtamoottorien ohjain ei ole täydellinen tai valmis, mutta se on hyvä ensimmäinen prototyyppi. Kommutoinnin ajoitus on epätasallinen ainakin sinimuotoista aaltomuotoa käyttävillä moottoreilla, ja laite on nykyisellä piirilevyllä hieman liian suuri lennökkikäyttöön. Testimoottoreita saatiin kuitenkin ohjattua halutulla tavalla, ja saavutettu maksiminopeus on huomattavan suuri. Työssä saatiin myös suunniteltua laitteen pääkohdat siten, että jatkokehitys on helppoa. Tehoaste on hyvin toimiva, samoin apujännitteiden muodostus ja vaihejännitteiden tarkkailun komparaattorikytkentä. Alkuperäisten tavoitteiden mukaisesti ohjaimella saatiin aikaiseksi luotettava käynnistyminen ja riippumattomuus moottorin parametreista. Jälkimmäinen on tosin hieman epävarma, sillä testejä ei päästy ajamaan varsinaisella trapetsoidaalisen jännitemuodon moottorilla, jotta oltaisiin voitu vertailla ajoituksen ja kommutointiennakon suhdetta ajettavaan moottorityyppiin.

Hyvin suoritettun jatkokehityksen jälkeen rakennettava toinen prototyyppi saattaa jo olla niin luotettava, että sitä uskalletaan alustavan testauksen jälkeen testata myös lopullisissa käyttöolosuhteissa lennokin sisällä. Kun ohjain läpäisee nämä testit, voidaan ohjaimen dokumentaatio julkaista Internetissä, jotta muut harrastajat voivat rakentaa omat versionsa laitteesta ja mahdollisesti jatkokehittää sitä.

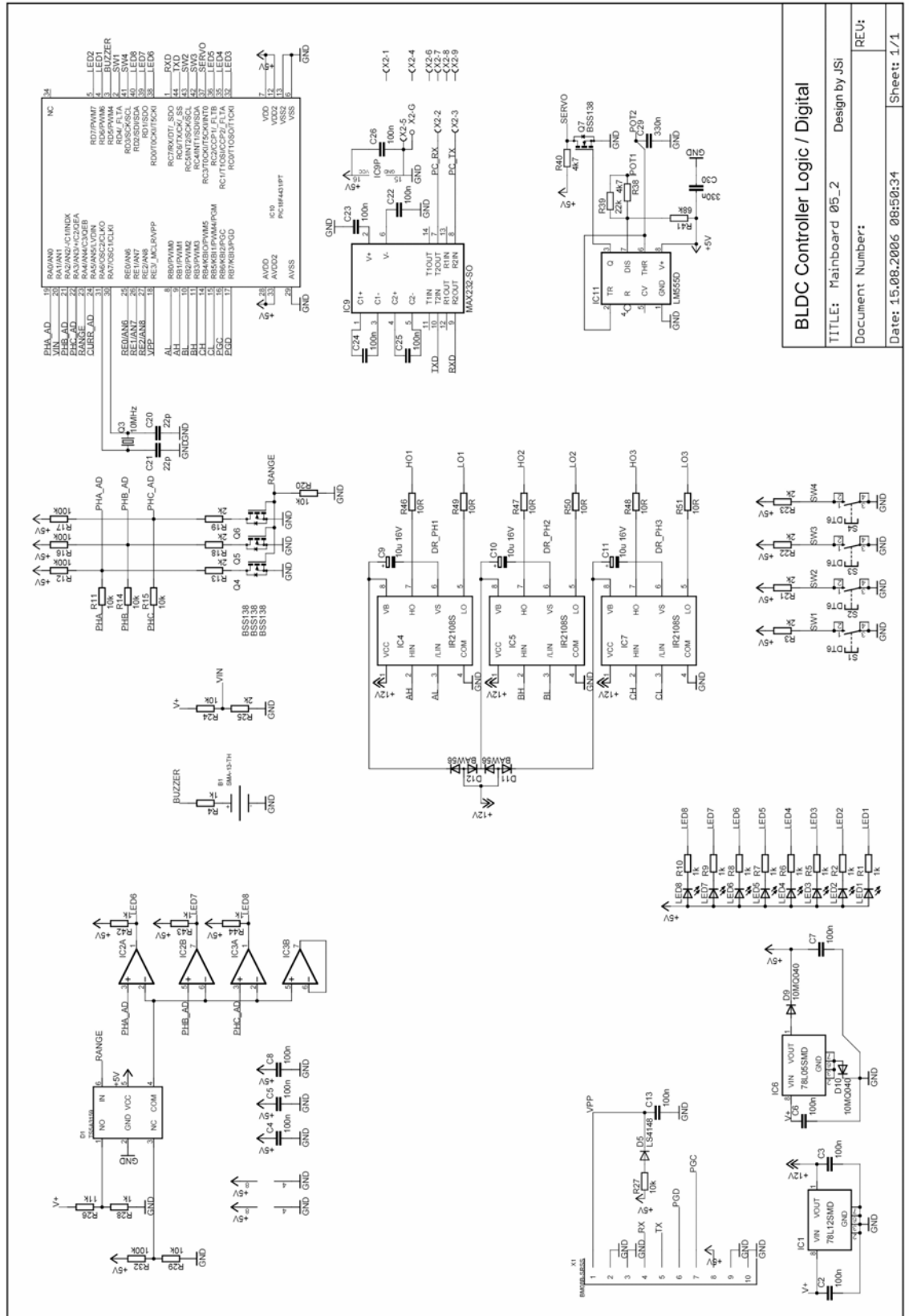
Lähdeviitteet

- ¹ P.Zielinski, K.Schoepp: Three-phase Low-speed Permanent Magnet Synchronous Machines, Institute of Electrical Machine Systems, Technical University of Wroclaw, Wroclaw 1991, P. Rotherin mukaan, http://www.torcman.de/peterslrk/Paper_from%20_Wroclaw.html, viitattu 31.7.2006.
- ² L. Retzbach: Torquemax LRK sensor and brushless do-it-yourself motor, P. Rotherin mukaan, http://www.aerodesign.de/peter/2001/LRK350/EM-4-2000/LRK-EM-4-2000_eng.htm, viitattu 31.7.2006.
- ³ J. Luomi: Sähkökäyttöjen suunnittelu. Opintojakson S-81.320 opetusmoniste, TKK Sähkö- ja tietoliikennetekniikan osasto, 2004.
- ⁴ J. Luomi: Sähkökäyttöjen ohjaus. Opintojakson S-81.310 opetusmoniste, TKK Sähkö- ja tietoliikennetekniikan osasto, 2003.
- ⁵ S. Ogasawara, H Akagi: An Approach to Position Sensorless Drive for Brushless dc Motors, IEEE Transactions on Industrial Applications Vol 27 nro 5 s. 928-933, IEEE Press, New York 1991.
- ⁶ J. Shao: Direct Back EMF Detection Method for Sensorless Brushless DC (BLDC) Motor Drives, Virginia Polytechnic Institute and the State University, Blacksburg 2003, <http://scholar.lib.vt.edu/theses/available/etd-09152003-171904/unrestricted/T.pdf>, viitattu 31.7.2006.
- ⁷ ST Microelectronics: <http://www.st.com/stonline/products/promlit/pdf/brst7mc-1104.pdf>, viitattu 28.9.2006
- ⁸ Fairchild: <http://www.fairchildsemi.com>, viitattu 31.7.2006
- ⁹ International Rectifier: <http://www.irf.com/technical-info/appnotes/an-937.pdf>, viitattu 31.7.2006.
- ¹⁰ I. Lindell, A Sihvola: Sähkömagneettinen kenttäteoria osa 1 Staattiset kentät s. 164-168, Otatieto / Oy Yliopistokustannus, Helsinki 2001.
- ¹¹ Allegro: <http://www.allegromicro.com/>, viitattu 31.7.2006.
- ¹² Sentron: <http://www.sentron.ch/>, viitattu 31.7.2006.
- ¹³ Modulohm: <http://www.modulohm.dk/serie-1.html>, viitattu 31.7.2006.
- ¹⁴ Welwyn: http://www.welwyn-tt.co.uk/pdf/brochures/ShortFormA4_IssueF.pdf, viitattu 31.7.2006.
- ¹⁵ Epcos: http://www.epcos.com/inf/20/30/db/alu_03/03150323.pdf, viitattu 31.7.2006.
- ¹⁶ Nichicon: <http://www.nichicon.com/english/seihin/pdfs/e-pj.pdf>, viitattu 31.7.2006.
- ¹⁷ International Rectifier: <http://www.irf.com/product-info/datasheets/data/ir2108.pdf>, viitattu 31.7.2006.
- ¹⁸ National Semiconductor: <http://cache.national.com/ds/LM/LM5100.pdf>, viitattu 31.7.2006.
- ¹⁹ National Semiconductor: <http://cache.national.com/ds/LM/LM78L05.pdf>, viitattu 31.7.2006.
- ²⁰ Texas Instruments: <http://focus.ti.com/lit/ds/symlink/tps5430.pdf>, viitattu 31.7.2006.

LIITE 1a



LIITE 1b



LIITE 2

```
#include "D:\projects\BLDC\18f_1.h"

#use fast_io(A)
#use fast_io(B)
#use fast_io(C)
#use fast_io(D)
#use fast_io(E)

short servopol=1;      //1 = positiivinen pulssi aktiivinen, 0 = negatiivinen
short edge0;           //1 = servopulssin alkureuna tulossa, 0 = loppureuna
short servovalue=0;    //Saatu servopulssin alku- ja loppuaika
short running=0;       //Ajo päällä
long pwmduty=0;        //Pulssisuhde
int servowd=0;         //Vahtii että servopulsseja tulee
int badservo=0;        //Vahtii servopulssien oikeaa pituutta
int phase=0;           //0=freewheel, 1=AB, 2=AC, 3=BC, 4=BA, 5=CA, 6=CB, 7=brake
int COMP[7]={0,2,1,0,2,1,0}; //Määrittää komparaattorin jota jaksolla
                                //tarkkaillaan
int DCH[8]={0,0,0,1,1,2,2,0}; //Pulssisuhteen kanavan valinta
long OVD[8]={0,0xC304,0xC310,0xCC10,0xCC01,0xF001,0xF004,0x0015};
                                //Määrittää ohitetut teholähdöt ja niiden tilat
int IND[8]={0x07,0x44,0x42,0x41,0x44,0x42,0x41,0x00};
                                //Määrittää minkä kanavien ylä- ja alakytkimet
                                //ovat toisistaan riippumattomia
                                //(kaikki,C,B,A,C,B,A,ei mitään)

long servocurrent=0;     //Ajantasainen servoarvo
long servostart;         //Servopulssin alkuhetki
long servoend;           //Servopulssin loppuhetki
short COMM=0;           //Timer1 käskee tällä kommutoimaan
long ZTOC;               //Aika nollan ylityksestä kommutointiin
long CTOZ=16000;         //Aika kommutoinnista nollan ylitykseen
long limit=16000;        //Maksimiaika kommutoinnista nollan ylitykseen
long CURR[4]={0,0,0,0}; //Virta-arvojen bufferi
long curragv=0;          //Liukuva keskiarvo neljästä viimeisestä virta-arvosta
long currlimit=800;      //Maksimivirta
int buffD;               //Bufferi I/O-portti D:n hetkelliselle arvolle
int buffB;               //Bufferi I/O-portti B:n hetkelliselle arvolle

#int_TIMER0              //Timer0 vuotaa yli 6,5536ms välein
TIMER0_isr()
{
    servowd++;
    if(servowd>=100) servocurrent=0; //0,66s paussi servosignaalisissa sammuttaa
                                    //moottorin
}
```

```

#int_TIMER1                //Timer1 vuotaisi yli 26,2144ms välein jos ajosilmukka
                            //ei muuttaisi laskurin arvoa kommutoinnin mukaan

TIMER1_isr()
{
    COMM=1;                //Ohjaa ohjelman suoritus ajosilmukasta kommutointiin
}

#int_TIMER2                //Timer2 vuotaa yli 400us välein
TIMER2_isr()
{
    if(servocurrent>pwmduty) pwmduty++; //Ohjaus tasainen ramppi, laidasta lai-
    if(servocurrent<pwmduty) pwmduty--; //taan 2000*400us=0,8s
    if(curravg>currlimit) pwmduty-=3;  //Ylivirta laskee pulssisuhdetta nopeasti
    if(servocurrent==0) running=0;      //Jos kaasu vedetään kokonaan pois,
                                        //pyydetään kommutointirutiinia sammutta-
                                        //maan ajo välittömästi
}

#int_EXT                    //Servopulssin keskeytys
EXT_isr()
{
    if(!servovalue) //Jos aiempaa mittausta ei ole käsitelty, ei mitata uutta
    {
        if (edge0==servopol)          //Jos on nouseva reuna...
        {
            servostart=get_timer0();    //Merkitään alkuaika servostartiin
            servowd=0;                  //Potkaistaan servon vahtipiskiä
            if(servopol) Ext_int_edge(H_TO_L);
            else ext_int_edge(L_TO_H);   //Jäädään odottamaan laskevaa reunaa
            edge0=!servopol;
        }
        else                          //Onkin laskeva reuna, joten...
        {
            servoend=get_timer0();      //Loppuaika servoendiin
            servovalue=1;                //Merkitään mittausta saaduksi
            if(servopol) Ext_int_edge(L_TO_H);
            else ext_int_edge(H_TO_L);   //Seuraavaksi havaitaan nouseva reuna
            edge0=servopol;
        }
    }
}

#int_AD                    //A/D-muunnos valmis
AD_isr()
{
    curr[3]=curr[2];                //Pyöritetään uusi tulos FIFO-bufferiin
    curr[2]=curr[1];
}

```

```

    curr[1]=curr[0];
    curr[0]=ADRESULT;
    curragv=(curr[0]+curr[1]+curr[2]+curr[3])>>2; //Lasketaan liukuva keskiarvo
}

long servocalc(long start, long stop) //Funktio pulssin pituuden laskemiselle
{
    long servoc;
    servoc=stop-start;
    if ((servoc<=8000)|| (servoc>=22000)) //Alle 0,8ms/yli 2,2ms on häiriö
    {
        badservo++;
        if(badservo>=8) servoc=0; //Kahdeksan häiriöpulssia
    } //aiheuttaa ajon keskeyttämisen
    else
    {
        if(servoc<=11000) servoc=11000;
        else if(servoc>=19000) servoc=19000;
        servoc-=11000; //Vähennetään minimipulssin kesto
        servoc=servoc>>2; //Skaalataan PWM-alueelle 0-2000
        badservo=0; //Merkitään hyvä pulssi
    }
    return(servoc);
}

void main()
{
    phase=0; //Nollajakso, ohjaukset poissa päältä
    OVDCON=OVD[phase];
    PWMCON0=IND[phase];
    PWMCON=PWM_0to5|PWM_OD_SYNC; //Kuusi lähtöä komplementaarisina
    output_low(RANGE); //Pienen pulssisuhteen toiminta-alue
    set_tris_A(0x2F); //Määritetään I/O-lähtöjen toimintasunnat
    set_tris_B(0xC0);
    set_tris_C(0xBC);
    set_tris_D(0x1F);
    set_tris_E(0xFF);
    PORTD=0xFF; //Asetetaan lähdöt inaktiivisiksi
    Output_low(buzzer);
    output_low(LED1);
    output_low(LED2);
    PORTC=0;
    delay_ms(200);
    output_high(LED1);
    output_high(LED2);
    PORTC=0xFF;
    delay_ms(200);
}

```

```

printf("\n\rBLDC controller v5.121 5.6.2006\r\n");

//Lähetetään sarjaportin kautta versiotieto
setup_spi(FALSE); //SPI-väylää ei käytetä
setup_wdt(WDT_OFF); //Sisäistä vahtikoiraa ei käytetä
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1); //Timer0:aa kasvatetaan 250ns välein
setup_timer_1(T1_INTERNAL|T1_DIV_BY_4); //Timer1:ä kasvatetaan 1us välein
setup_timer_2(T2_DIV_BY_16,249,1); //Timer2:a kasvatetaan 4us välein,
//ja laskuri nollautuu 249:n jälkeen

enable_interrupts(INT_TIMER0); //Aktivoidaan käytettävät keskeytykset
enable_interrupts(INT_TIMER1);
enable_interrupts(INT_TIMER2);
enable_interrupts(INT_EXT);
enable_interrupts(INT_AD);
enable_interrupts(GLOBAL);

PTCON=PWM_CLOCK_DIV_4|PWM_FREE_RUN|PWM_COUNT_UP;
set_PWM_period(499); //Alustetaan PWM-osio: taaajuus 20kHz, säätöalue 0-2000

ADCON1=INT_VREFS; //Alustetaan A/D-muunnin:
//TAD=1,6us, TACQ=32us, Tconv=121,6us
ADCON2=AD_RIGHT_JUSTIFIED|T_ACQ_64TAD|AD_CLK_FOSC_DIV_32;
ADCON3=AD_INT_EACH_WORD;
ADCHS=GROUP_A_AN0|GROUP_C_AN2|GROUP_D_AN3|GROUP_B_AN5;
ANSEL=AN0|AN1|AN2|AN3|AN5;
ADCON0=CONT_LOOP|SINGLE_CH_MODE_2|GO_NOTDONE|AD_ENABLED;
delay_us(50);

Servo0: //Odotetaan kaksi servomittausta ennen kuin jatketaan
while(!servovalue);
servocurrent=servocalc(servostart, servoend);
servovalue=0;
while(!servovalue);
servocurrent=servocalc(servostart, servoend);
servovalue=0;
if(servocurrent!=0) goto Servo0; //Ohjauksen on oltava poissa päältä

Servo1:
servowd=0;
while(!servovalue);
servocurrent=servocalc(servostart, servoend);
servovalue=0;
servowd=0;
delay_ms(20);
servowd=0;
delay_ms(20);
servowd=0;
if(servocurrent<=100) goto Servo1; //Jatketaan vasta kun tarpeeksi ohjausta

```

```

Start:
    PORTB=0x55;                //Asetetaan lähdöt inaktiivisiksi
    PORTC=0xFF;
    PORTD=0xFF;
    Output_low(buzzer);
    Output_low(range);
    pwmduty=100;                //Asetetaan minimipulssisuhde
    phase=1;                    //Asetetaan ensimmäinen jakso aktiiviseksi
    OVDCON=OVD[phase];
    PWMCON0=IND[phase];
    set_PWM_duty(DCH[phase],pwmduty); //Päivitetään pulssisuhde jaksolle
    set_timer1(0);              //Nollataan timer1 jakson mittausta varten
    running=1;                  //Merkittään ajo alkaneeksi

Driveloop_low:                  //Alhaisen pulssisuhteen ajosilmukka
    if (servovalue)              //Käsitellään mahdollinen servomittaus
    {
        servocurrent=servocalc(servostart, servoend);
        servovalue=0;            //Merkittään mittaus käsitellyksi
    }
    set_PWM_duty(DCH[phase],pwmduty); //Päivitetään pulssisuhde jaksolle

WaitZC_low:                     //Alhaisen pulssisuhteen nollakohdan odotus
    if(get_timer1()>limit)       //Jos nollan ylitystä ei tule ajoissa...
    {
        CTOZ=limit;
        goto Secondhalf_high;    //...odotetaan vielä lisää ja kommutoidaan
    }                             //sitten seuraavaan jaksoon
    buffB=input_B();              //Otetaan PWM-tilat ja komparaattoreiden
    buffD=input_D();              //lähdöt talteen jotteivät muutu vertailun aikana
    if((bit_test(phase,0)==bit_test(buffD,COMP[phase]))|(!buffB&0x1A))
    {
        goto waitZC_low;         //Jos nolla ei ole ylittynyt, jatketaan odottelua
    }
    CTOZ=get_timer1();            //Otetaan talteen aika kommutoinnista nollan
                                //ylitykseen
    if((INPUT_B()&0x1A)) goto WaitZC_low; //Ylitys olikin PWM-pulssi, ohitetaan
    delay_us(3);
    if(bit_test(phase,0)==bit_test(INPUT_D(),COMP[phase])) goto WaitZC_low;
                                //Tarkistetaan vielä että komparaattori vaihtoi
                                //oikeasti tilaansa

Secondhalf_low:                 //Tässä asetetaan timer1 vuotamaan yli kun on aika
    ZTOC=(CTOZ);                 //kommutoida
    if(ZTOC<80) set_timer1(65533);

```

```

else set_timer1(78-ZTOC); //78 korjaa ohjelman suorituksesta tulevat viiveet:
                        //24 ZC->set_timer, 54 set_timer->comm (keskimäärin)

Waitcomm_low:          //Odotetaan lupaa kommutoida
    if(COMM) goto Commutation_low;
    if (servovalue)      //Käsitellään tarpeen vaatiessa servomittaus
    {
        servocurrent=servocalc(servostart, servoend);
        servovalue=0;
    }
    set_PWM_duty(DCH[phase],pwmduty);    //Päivitetään pulssisuhde
    if(COMM) goto Commutation_low;
    else goto Waitcomm_low;

Commutation_low:
    phase++;                //Vaihdetaan jakso seuraavaan
    if(phase>=7) phase=1;
    set_PWM_duty(DCH[phase],pwmduty);    //Päivitetään pulssisuhde
    OVDCON=OVD[phase];
//Päivitetään jaksotieto tehoasteelle
    PWMCON0=IND[phase];
    set_timer1(0);
                        //Aloitetaan mittaus kommutoinnista
                        //nollan ylitykseen
    delay_us(50);          //Odotetaan induktiivisen jännitepulssin
                        //maksimikesto
    if(!running) goto turnoff;    //Lopetetaan ajo jos ohjaus on nolla
    COMM=0;                //Valmistaudutaan seuraavaan kommutointi-
                        //käskyyn
    if(pwmduty>1099)        //Vaihdetaan tarpeen vaatiessa toiminta-
    {                        //aluetta
        output_high(RANGE);
        goto driveloop_high;
    }
    goto driveloop_low;      //Siirrytään takaisin ajosilmukkaan

Driveloop_high:          //Korkean pulssisuhteen ajosilmukka, toiminnaltaan
                        //yhteneväinen matalan pulssisuhteen ajosilmukkaan
    if (servovalue)
    {
        servocurrent=servocalc(servostart, servoend);
        servovalue=0;
    }
    set_PWM_duty(DCH[phase],pwmduty);

WaitZC_high:
    if(get_timer1())>limit)

```

```

    {
        CTOZ=0;
        goto Secondhalf_high;
    }
    buffB=input_B();
    buffD=input_D();
    if((bit_test(phase,0)==bit_test(buffD,COMP[phase]))||(!buffB&0x1A))
    {
        goto waitZC_high;
    }
    CTOZ=get_timer1();
    delay_us(3);
    if(bit_test(phase,0)==bit_test(INPUT_D(),COMP[phase])) goto WaitZC_high;

Secondhalf_high:
    ZTOC=(CTOZ);
    if(ZTOC<100) set_timer1(65530);
    else set_timer1(95-ZTOC);

Waitcomm_high:
    if(COMM) goto Commutation_high;
    if (servovalue)
    {
        servocurrent=servocalc(servostart, servoend);
        servovalue=0;
    }
    set_PWM_duty(DCH[phase],pwmduty);
    if(COMM) goto Commutation_high;
    else goto Waitcomm_high;

Commutation_high:
    phase++;
    if(phase>=7) phase=1;
    set_PWM_duty(DCH[phase],pwmduty);
    OVDCON=OVD[phase];
    PWMCON0=IND[phase];
    set_timer1(0);

    if(pwmduty<901)
    {
        output_low(RANGE);
        delay_us((1300-pwmduty)>>4);
        delay_us(100);
        COMM=0;
        goto driveloop_low;
    }

    delay_us(10);

```

```
    if(!running) goto turnoff;
    COMM=0;
    goto driveloop_high;

turnoff:                //Ajon lopetus
    phase=0;            //Kaikki tehoasteen lähdöt inaktiivisiksi
    OVDCON=OVD[phase];
    PWMCON0=IND[phase];
    goto servo0;        //Siirrytään odottamaan seuraavaa käynnistystä
}
```